



AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許
(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB,
GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR),
OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

規則4.17に規定する申立て:

- すべての指定国のための不利にならない開示又は新規性喪失の例外に関する申立て (規則4.17(v))

添付公開書類:

- 国際調査報告書
- 不利にならない開示又は新規性喪失の例外に関する申立て

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明細書

疑似乱数発生方法及び疑似乱数発生器

〔発明の属する技術分野〕

- 5 本発明は、暗号通信やデジタル署名などで利用する疑似乱数を発生させる疑似乱数発生方法や、疑似乱数発生器、乱数発生プログラムに関する。

〔従来の技術〕

- 10 従来より、有線や無線により情報通信を行う際に、内容が第三者に漏れないように情報を暗号化して送信することが行われている。この暗号化方式の一つに逐次暗号方式（ストリーム暗号方式）がある。逐次暗号方式は、送信側と受信側で同一の疑似乱数を発生させ、送信側は疑似乱数のビット列と平文のビット列とを用いて暗号文のビット列を作成し暗号文として受信側に送出し、受信側は送信側から受信した暗号文のビット列と疑似乱数のビット列とを用いて平文のビット列を求め平文に復号化するものである。

- 15 第16図は、従来の逐次暗号方式を説明する図である。送信側の暗号化装置100は、疑似乱数発生器101と論理演算処理部102を有しており、受信側の復号化装置110は、疑似乱数発生器111と論理演算処理部112を有している。

- 20 暗号化装置100の疑似乱数発生器101と復号化装置110の疑似乱数発生器111は、同一の秘密鍵を与えることによって互いに全く同一の疑似乱数を発生する論理構造を有している。また、暗号化装置100の論理演算処理部102と復号化装置110の論理演算処理部112は、ビット単位で排他的論理和の演算処理を行う。

- 25 第17図は、暗号化装置100の疑似乱数発生器101を説明する図である。尚、復号化装置110の疑似乱数発生器111については、暗号化装置100の疑似乱数発生器101と同一の構成を有するのでその詳細な説明を省略する。

疑似乱数発生器101は、非線形コンバイナ型の疑似乱数発生器（Nonlinear Combiner Generator）であり、第17図に示すように、並列に配置した複数の線形フィードバックシフトレジスタ（Linear Feedback Shift Register; LFSR）10

3と、非線形変換部104とを有しており、各線形フィードバックシフトレジスタ103から出力したビット列を非線形変換して疑似乱数を発生させる。本従来例では、各線形フィードバックシフトレジスタ103は、1回のシフト動作でそれぞれ1ビット(X_1 、 X_2 、 \dots 、 X_L)を出力し、非線形変換部104は、各
5 線形フィードバックシフトレジスタ103から入力されたビット列をもとに1ビットの疑似乱数を出力する構成を有している。

第18図は、一般的な線形フィードバックシフトレジスタ103の構成を簡単に説明する図である。線形フィードバックシフトレジスタ103は、1ビットの情報を記憶できる複数のシフトレジスタ105と、複数の排他的論理和演算回路
10 106とを有し、各シフトレジスタ105の出力と各排他的論理和演算回路106の一方の入力との間にはフィードバックタップ107が接続されている。フィードバックタップ107(c_{n-1} 、 c_{n-2} 、 \dots 、 c_n)は、1のとき結線を示し、0のとき断線を示し、それぞれが予め1または0に定められている。

このシフトレジスタ105の個数を n とすると、1つのシフトレジスタ105
15 に注目したとき、出力系列の最大周期は、 $(2^n) - 1$ となることが知られており、この系列をM系列という。(尚、「 2^n 」は、2の n 乗(2^n)を意味する。以下、指数部分は、その前に「 \wedge 」を付して示す。)

例えば、第14図に示す線形フィードバックシフトレジスタ103の場合、M系列を生成する特性多項式は、以下の式で表される。

20
$$C(x) = (X^n) + c_{n-1}(X^{(n-1)}) + \dots + c_1X + 1$$

上記の特性多項式で第1項目の指数 n は線形フィードバックシフトレジスタ103の次数、すなわちシフトレジスタの個数を示し、2項目以降の指数部分は、フィードバックタップによる結線位置を示している。上記の式に示す特性多項式が原始多項式となるようにすれば、線形フィードバックシフトレジスタは、M系
25 列を出力する。

このような従来の非線形コンバイナ型疑似乱数発生器は、ビット単位の論理演算を基にした簡単なロジックで構成できるので、いわゆるハードウェアでの実装に適していると考えられている。

尚、従来より、線形フィードバックシフトレジスタからの出力を排他的論理和

等の演算処理によって変更することが提案されている(例えば、特許文献1 参照。)

【特許文献1】

特開平6-342257号

【発明が解決しようとする課題】

5 (第1の解決課題)

しかしながら、線形フィードバックシフトレジスタ103は、シフトレジスタ数の2倍の出力を観測することで、線形フィードバックシフトレジスタ103の構成、すなわちシフトレジスタ数及び結線位置と、初期値の全てを特定することが可能である。したがって、構成が固定された線形フィードバックシフトレジスタ103をそのまま疑似乱数発生器101に用いるには、暗号強度が弱く、安全性に問題がある。

また、線形フィードバックシフトレジスタ103は、特性多項式の変更によりシフトレジスタの結線位置や結線数を変更すると、線形フィードバックシフトレジスタの出力がM系列ではなく短周期となって暗号強度が低下するおそれがあることから、特性多項式は予めM系列を出力する値に固定されており、線形フィードバックシフトレジスタの構成を簡単に変更することはできないと考えられている。

(第2の解決課題)

また、従来の非線形コンバイナ型疑似乱数発生器は、線形フィードバックシフトレジスタ103で1ビット単位の演算を連続して繰り返し実行しなければならない。このような処理は、ハードウェアでは得意とするところであり、比較的高速に処理できるが、ソフトウェアでは苦手とするものであり、ハードウェアの場合と比較して処理速度が極端に遅くなる。

一方、非線形変換部104は、論理積や排他的論理和などの単純な演算を実行している。したがって、線形フィードバックシフトレジスタ103のスループットが、非線形変換部104のスループットよりも下回り、発生器全体の中で乱数ビット列を出力する部分、すなわち線形フィードバックシフトレジスタ103がボトルネックになってしまう。このため、従来の非線形コンバイナ型疑似乱数発生器は、ソフトウェアで実装した場合にはハードウェアで実装した場合よりも全

体のスループットが低下するという問題があり、ソフトウェアでは使用することが困難であった。

また、疑似乱数の暗号強度を十分に確保するためには、線形フィードバックシフトレジスタ 103 のシフトレジスタ 105 の数や、線形フィードバックシフトレジスタ 103 の個数をある程度の数以上必要とする。しかし、スループットは、線形フィードバックシフトレジスタ 103 のシフトレジスタ 105 の数が増加するほど、或いは線形フィードバックシフトレジスタ 103 の個数が増加するほど低くなるという、相反する関係にある。したがって、高い暗号強度を確保しつつ、高いスループットを実現することは困難であった。

- 10 本発明は、上述の第 1 及び第 2 の解決課題の少なくとも一方を解決すべくなされたものであり、その目的は、強い暗号強度を維持しつつ線形フィードバックシフトレジスタの構成を容易かつ動的に変更することができ、また、十分に高い暗号強度を確保しつつ、より高いスループットを実現できる疑似乱数発生方法等を提供することにある。

15 [課題を解決するための手段]

- 上記課題を解決する請求項 1 に記載の発明による疑似乱数発生方法は、 n 個のシフトレジスタを有し、1 周期分のビット数が $(2^n) - 1$ 個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定する第 1 ステップと、所定の演算処理により初期値から線形フィードバックシフトレジスタの 1 周期分のビット数と互いに素である導出値を求める第 2 ステップと、導出値と線形フィードバックシフトレジスタの 1 周期分のビット数を 2 倍以上した値とを乗算して、第 1 線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出する第 3 ステップと、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させる第 4 ステップと、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する第 5 ステップと、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成する第 6 ステップと、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる第 7 ステップと、を有することを特徴とする。

この発明は、出力系列がM系列のビット列をs個ごとにサンプルしたビット列は、そのM系列の1周期分のビット数($= (2^n) - 1$)と導出値が互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタのM系列を構成し、また、少なくとも2周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用するものである。

この発明によると、n個のシフトレジスタを有し、1周期分のビット数が($2^n - 1$)個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定し、所定の演算処理により初期値から線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める。

そして、導出値と線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値とを乗算して、第1線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出し、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させ、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する。

そして、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成し、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる。

これによれば、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタからM系列のビット列を出力させることができる。したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

請求項2の発明は、請求項1に記載の疑似乱数発生方法において、初期値に対してハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数を導出値として採用することを特徴とする。

この発明によると、初期値に対してハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数を導出値として採用するので、導出値の推定困難度を高めることができ、より高度な秘匿性を得ることができる。

請求項3の発明は、請求項1または2に記載の疑似乱数発生方法において、線形フィードバックシフトレジスタの再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われることを特徴とする。

5 この発明は、少なくとも2周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができるという、バーレイキャンプマッセイアルゴリズムを利用するものである。

請求項4の発明は、請求項1～3のいずれかに記載の疑似乱数発生方法において、第6ステップの発明は、発生させた疑似乱数を非線形変換する第7ステップを有することを特徴とする。この発明によると、発生させた疑似乱数を非線形変換するので、疑似乱数に非線形性を与えることができ、暗号強度を更に向上させることができる。

請求項5に記載の発明による疑似乱数発生器は、 n 個のシフトレジスタを有し、1周期分のビット数が $(2^n) - 1$ 個となるビット列を出力可能な線形フィードバックシフトレジスタと、秘密鍵に基づき線形フィードバックシフトレジスタの初期値を設定する初期値設定手段と、所定の演算処理により初期値から線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める導出値算出手段と、線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値と導出値とを乗算して、第1線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出するビット数算出手段と、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させるビット列出力手段と、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する新ビット列生成手段と、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成する線形フィードバックシフトレジスタ再構成手段と、再構成後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる疑似乱数発生手段と、を有することを特徴とする。

この発明は、出力系列がM系列のビット列を s 個ごとにサンプルしたビット列は、そのM系列の1周期分のビット数 $(= (2^n) - 1)$ と導出値 s が互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタのM系

列を構成し、また、少なくとも2周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用するものである。

この発明によると、 n 個のシフトレジスタを有し、1周期分のビット数が $(2^n) - 1$ 個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定し、所定の演算処理により初期値から線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める。

そして、導出値と線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値とを乗算して、第1線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出し、その算出したビット数分のビット列を線形フィードバックシフトレジスタから初期値をもとに出力させ、その出力したビット列から導出値の間隔ごとにビット列を取り出して新ビット列を生成する。

そして、その新ビット列を出力可能な構成に線形フィードバックシフトレジスタの構成を再構成し、再構成した後の線形フィードバックシフトレジスタから初期値をもとに疑似乱数を発生させる。

これによれば、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタからM系列のビット列を出力させることができる。したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

請求項6に記載の発明は、請求項5に記載の疑似乱数発生器において、線形フィードバックシフトレジスタ再構成手段の代わりに、新ビット列を出力可能な構成を有する第2の線形フィードバックシフトレジスタを生成する線形フィードバックシフトレジスタ生成手段を設け、疑似乱数発生手段は、第2の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させることを特徴とする。この発明によると、線形フィードバックシフトレジスタを第1の線形フィードバックシフトレジスタと第2の線形フィードバックシフトレジスタの2つに分けることができ、より秘匿性の向上を図ることができる。

請求項 7 に記載の発明による疑似乱数発生器は、秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力する選択用乱数ビット列出力手段と、選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段と、を有することを特徴とする。

- 5 この発明によると、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きな
- 10 ビット数を有する増幅乱数ビット列を得ることができる。

- 15 したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生
- 20 器全体のスループットを高速化することができる。

請求項 8 に記載の発明は、請求項 7 に記載の疑似乱数発生器において、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行う乱数テーブル初期設定手段を有することを特徴とする。

- 25 この発明によると、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行うので、秘密鍵を変更するごとに乱数テーブル内の初期値を変更することができる。したがって、暗号強度を増大させることができる。

請求項 9 に記載の発明は、請求項 7 または 8 に記載の疑似乱数発生器において、

選択用乱数ビット列出力手段が複数設けられ、乱数テーブルが各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、増幅乱数ビット列選択手段が各選択用乱数ビット列出力手段から各々出力された各選択用乱数ビット列を用いて各選択用乱数ビット列出力手段ごとに対応する乱数テーブルをそれぞれ参照し、

5 各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、非線形変換手段が各増幅乱数ビット列選択手段によって各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする。

この発明によると、各選択用乱数ビット列出力手段からそれぞれ選択用乱数ビット列が出力され、各選択用乱数ビット列を用いて各乱数テーブルがそれぞれ参照され、その参照により各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換することによって疑似乱数を発生させるので、従来はボトルネックとなっていた乱数ビット列を出力する部分のスループットを向上させることができ、疑似乱数発生器全体のスループットを高速化することができる。

10

15

請求項 10 に記載の発明は、請求項 9 に記載の疑似乱数発生器において、各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、増幅乱数ビット列選択手段により各乱数テーブル内から選択された各増幅乱数ビット列を選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力

20

する排他的論理和演算処理手段を有することを特徴とする。

この発明によると、各乱数テーブルから選択された各増幅乱数ビット列が選択用乱数ビット列出力手段ごとに排他的論理和演算してから非線形変換手段に出力されるので、増幅乱数ビット列発生手段によって発生させた乱数ビット列をそのまま用いたものよりも、暗号強度を増大させることができる。

25 請求項 11 に記載の発明は、請求項 9 または 10 に記載の疑似乱数発生器において、所定のタイミングで各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段を有することを特徴とする。

この発明によると、所定のタイミングで各乱数テーブル同士の入れ替えを行うので、参照元となる乱数テーブルを変更することができる。したがって、固定さ

れたものよりも暗号強度を増大させることができる。

請求項 1 2 に記載の発明は、請求項 1 1 に記載の疑似乱数発生器において、乱数テーブル入れ替え手段が、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに、各乱数テーブル同士

5 の入れ替えを行うことを特徴とする。

この発明は、上述の請求項に記載した所定のタイミングの一具体例を示したものである。これによると、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに各乱数テーブル同士の入れ替えを行う。したがって、短いサイクルで参照元となる乱数テーブルを変更す

10 ることができ、暗号強度を更に増大させることができる。

請求項 1 3 に記載の発明は、請求項 1 1 または 1 2 に記載の疑似乱数発生器において、乱数テーブル入れ替え手段は、各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、乱数テーブル入れ替え用乱数を乱数テーブルのテーブル番号として各乱数テーブルに付与し、テーブル番号をもとに予め

15 設定された規則に従って乱数テーブルの順番を入れ替えることを特徴とする。

この発明は、上述の乱数テーブル入れ替え手段についての具体的な一例を示したものである。これによると、乱数テーブル入れ替え用乱数を発生させ、乱数テーブルのテーブル番号として各乱数テーブルに付与し、その付与したテーブル番号をもとに乱数テーブルの順番を入れ替える。したがって、乱数テーブルの順番

20 を簡単かつ迅速に入れ替えることができ、非線形変換手段よりも上流側のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項 1 4 に記載の発明は、コンピュータを以下の手段として機能させるための疑似乱数発生プログラムであり、秘密鍵に基づいて所定ビット数を有する選択

25 用乱数ビット列を出力させる選択用乱数ビット列出力手段と、選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選

択手段と、増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段として機能させるためのプログラムである。

この発明によると、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きなビット数を有する増幅乱数ビット列を得ることができる。

したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来は、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換部のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

請求項 15 に記載の発明は、請求項 14 に記載の疑似乱数発生プログラムにおいて、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行う乱数テーブル初期設定手段としてコンピュータを機能させることを特徴とする。

この発明によると、秘密鍵が与えられることにより秘密鍵に基づいて増幅乱数ビット列を発生させ、乱数テーブルに格納して、乱数テーブルの初期設定を行うので、秘密鍵を変更するごとに乱数テーブル内の初期値を変更することができる。したがって、暗号強度を増大させることができる。

請求項 16 に記載の発明は、請求項 14 または 15 に記載の疑似乱数発生プログラムにおいて、選択用乱数ビット列出力手段は、複数設けられ、乱数テーブルは、各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、増幅乱数ビット列選択手段は、各選択用乱数ビット列出力手段から各々出力された各選択用乱数ビット列を用いて各選択用乱数ビット列出力手段ごとに対応する乱数テーブルをそれぞれ参照し、各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、非線形変換手段は、各増幅乱数ビット列選択手段によって各乱数

テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする。

この発明によると、各選択用乱数ビット列出力手段からそれぞれ選択用乱数ビット列が出力され、各選択用乱数ビット列を用いて各乱数テーブルがそれぞれ参照され、その参照により各乱数テーブルから選択された各増幅乱数ビット列を用いて非線形関数により非線形変換することによって疑似乱数を発生させるので、従来はボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

- 10 請求項 17 に記載の発明は、請求項 16 に記載の疑似乱数発生プログラムにおいて、各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、増幅乱数ビット列選択手段により各乱数テーブル内から選択された各増幅乱数ビット列を選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段としてコンピュータを機能させることを
- 15 特徴とする。

この発明によると、各乱数テーブルから選択された各増幅乱数ビット列が選択用乱数ビット列出力手段ごとに排他的論理和演算してから非線形変換手段に出力されるので、増幅乱数ビット列発生手段によって発生させた乱数ビット列をそのまま用いたものよりも、暗号強度を増大させることができる。

- 20 請求項 18 に記載の発明は、請求項 16 または 17 に記載の疑似乱数発生プログラムにおいて、所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段としてコンピュータを機能させることを特徴とする。

この発明によると、所定のタイミングで各乱数テーブル同士の入れ替えを行うので、参照元となる乱数テーブルを変更することができる。したがって、固定されたものよりも暗号強度を増大させることができる。

- 25 請求項 19 に記載の発明は、請求項 18 に記載の疑似乱数発生プログラムにおいて、乱数テーブル入れ替え手段は、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに、各乱数テーブル同士の入れ替えを行うことを特徴とする。

この発明は、上述の請求項に記載した所定のタイミングの一具体例を示したものである。これによると、各乱数テーブルを参照するために必要な選択用乱数ビット列を選択用乱数ビット列出力手段が出力するごとに各乱数テーブル同士の入れ替えを行う。したがって、短いサイクルで参照元となる乱数テーブルを変更することができ、暗号強度を更に増大させることができる。

請求項 20 に記載の発明は、請求項 18 または 19 に記載の疑似乱数発生プログラムにおいて、乱数テーブル入れ替え手段は、各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、乱数テーブル入れ替え用乱数を乱数テーブルのテーブル番号として各乱数テーブルに付与し、テーブル番号をもとに予め設定された規則に従って乱数テーブルの順番を入れ替えることを特徴とする。

この発明は、上述の乱数テーブル入れ替え手段についての具体的な一例を示したものである。これによると、乱数テーブル入れ替え用乱数を発生させ、乱数テーブルのテーブル番号として各乱数テーブルに付与し、その付与したテーブル番号をもとに乱数テーブルの順番を入れ替える。したがって、乱数テーブルの順番を簡単かつ迅速に入れ替えることができ、非線形変換手段よりも上流側のスループットを向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

[発明の実施の形態]

20 (第 1 の実施の形態)

次に、本発明の第 1 の実施の形態について図に基づいて説明する。

第 1 図は、本実施の形態における疑似乱数発生器 1 を説明する図である。本実施の形態では、非線形コンバイナ型の疑似乱数発生器 1 を例に説明する。

疑似乱数発生器 1 は、利用者から与えられる秘密鍵に基づいて初期値を設定する初期値設定部（図示せず）と、初期値設定部から受け取った初期値をもとに疑似乱数を生成する複数の疑似乱数生成部 10 と、これら複数の疑似乱数生成部 10 の出力側に各々接続され、各疑似乱数生成部 10 から出力される疑似乱数を非線形変換する非線形変換部 20 を有している。

初期値設定部は、利用者から与えられる秘密鍵をビット列に変換し、疑似乱数

生成部 10 の数に分割して、後述する疑似乱数生成部 10 の線形フィードバックシフトレジスタ 11 にそれぞれ割り当てる初期値を生成する処理を行う。

疑似乱数生成部 10 は、L 個が互いに並列に配置されており、それぞれ線形フィードバックシフトレジスタ 11 と、線形フィードバックシフトレジスタ再構成手段 12 を有している。

線形フィードバックシフトレジスタ 11 は、従来技術で説明したものと同様に、1 ビットの情報を記憶できる n 個のシフトレジスタと、排他的論理和演算回路を有している。そして、本実施の形態では、1 周期分のビット数 m が $(2^n) - 1$ 個となるビット列、いわゆる M 系列を出力可能な構成に予め設定されている。

第 2 図は、本実施の形態における線形フィードバックシフトレジスタ 11 の初期多項式を例示するものである。初期多項式は、M 系列を出力するように予め設定されている特性多項式であり、1 項目の指数部（第 2 図では「 \wedge 」で表している）がシフトレジスタの個数を示し、2 項目以降の指数部が排他的論理和演算回路に接続された結線位置を示している。例えば、1 段目の線形フィードバックシフトレジスタ 11 (LFSR 1) は、131 個のシフトレジスタを有し、8 番目、3 番目、2 番目のシフトレジスタがフィードバックタップによって排他的論理和演算回路に接続されていることを示している。尚、本実施の形態では、シフトレジスタの個数 n は、全て素数個に設定されている。

線形フィードバックシフトレジスタ再構成手段 12 は、線形フィードバックシフトレジスタ 11 の構成を秘密鍵によって動的に変更して再構成するものである。具体的には、出力系列が M 系列のビット列を s 個ごとにサンプルした新ビット列は、M 系列の 1 周期分のビット数 $m (= (2^n) - 1)$ と導出値 s とが互いに素であるとき、すなわち、1 以外の共通の約数を持たないときは、他の構成を有する線形フィードバックシフトレジスタの M 系列になり、また、バーレイキャンブマッセイアルゴリズムによって、少なくとも 2 周期分以上のビット数を有するビット列から、そのビット列を出力可能な等価で最小の線形フィードバックシフトレジスタの特性多項式を求めることができることを利用して、線形フィードバックシフトレジスタ 11 の再構成を行う。

線形フィードバックシフトレジスタ再構成手段 12 は、初期値設定部によって

与えられた初期値から導出値 s を算出し、導出値 s と線形フィードバックシフトレジスタ 11 の 1 周期分のビット数 m ($= (2^n) - 1$) を 2 倍した値 $2m$ とを乗算し、線形フィードバックシフトレジスタ 11 から出力させるビット列のビット数 $2ms$ を算出する。

- 5 そして、初期値をもとに線形フィードバックシフトレジスタ 11 から $2ms$ 個のビット列を出力させ、その $2ms$ 個のビット列から導出値 s の間隔ごとにビット列を取り出して新ビット列を生成し、その新ビット列を用いてバーレイキャンブマッセイアルゴリズムにより線形フィードバックシフトレジスタ 11 の構成を変更する。

- 10 尚、本実施の形態では、線形フィードバックシフトレジスタ 11 から出力させるビット列のビット数が $2ms$ 個である場合を例に説明しているが、新ビット列のビット数が $2m$ 個以上であれば、等価な最小の線形フィードバックシフトレジスタを求めることができるので、 $2ms$ 個以上であればよい。

- バーレイキャンブマッセイアルゴリズムとは、線形フィードバックシフトレジスタ 11 のシフトレジスタの個数 n (線形複雑度) の 2 倍以上のビット数を有するビット列を入手することで、そのビット列を出力可能な等価な最小の線形フィードバックシフトレジスタを得ることができるというアルゴリズムである。バーレイキャンブマッセイアルゴリズムについては、例えば、文献 1 「暗号理論入門 (第 2 版)」、共立出版社、岡本栄司著、2002 年 4 月 10 日発行、に詳細に
- 15 説明されている。
- 20 次に、上記構成を有する疑似乱数発生器 1 の動作について第 3 図のフローチャートを用いて以下に説明する。

- まず最初に、初期値設定部によって初期値が設定される (ステップ S1)。初期値は、利用者から与えられる秘密鍵を所定の演算処理によって分割することによって設定される。
- 25 例えば、秘密鍵の長さが 16 バイトで「ABCDEFGHIJKLMN OP」であり、疑似乱数生成部 10 が 8 段の場合には、初期値は下記のように設定される。

例えば、秘密鍵の長さが 16 バイトで「ABCDEFGHIJKLMN OP」であり、疑似乱数生成部 10 が 8 段の場合には、初期値は下記のように設定される。

LFSR1 AB+X' FF' 埋め込み文字 (Padding)

LFSR2 CD+X' FF' 埋め込み文字 (Padding)

LFSR3 EF+X' FF' 埋め込み文字 (Padding)

LFSR4 GH+X' FF' 埋め込み文字 (Padding)

LFSR5 IJ+X' FF' 埋め込み文字 (Padding)

5 LFSR6 KL+X' FF' 埋め込み文字 (Padding)

LFSR7 MN+X' FF' 埋め込み文字 (Padding)

LFSR8 OP+X' FF' 埋め込み文字 (Padding)

ここでは、初期値は、秘密鍵「ABCDEFGHIJKLMNOP」を、「AB」、「CD」、・・・、「OP」の2文字ずつに分割し、残りのシフトレジスタを埋め込み文字 (Padding) で埋めることによって設定される。尚、上述の初期値設定方法は、実施例の一つであり、他の方法によって設定してもよい。

初期値設定部において秘密鍵から初期値が設定されると、各初期値は、各疑似乱数生成部10にそれぞれ入力され、線形フィードバックシフトレジスタ11のシフトレジスタ内にセットされる。

15 次に、線形フィードバックシフトレジスタ再構成手段12によって、線形フィードバックシフトレジスタ11の構成を再構成する処理が行われる (ステップS2～ステップS6)。

ここでは、まず、所定の演算処理により初期値から線形フィードバックシフトレジスタ11の1周期分のビット数 m と互いに素である導出値 s を算出する (ステップS2)。導出値 s は、初期値に対して、例えばMD5 (Message Digest 5) などのハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数が採用される。したがって、導出値の推定困難度を高めることができ、より高度な秘匿性を得ることができる。尚、導出値 s は、初期値から求めることができ、かつビット数 m と互いに素であればよく、上記の算出方法によって求められるものに限定されない。但し、秘匿性を維持するために、上記所定の演算処理は、
25 一方向性を満足しうる演算処理でなければならない。

導出値 s を算出すると、次に、線形フィードバックシフトレジスタ11から出力させるビット列のビット数 $2ms$ を算出する (ステップS3)。線形フィードバックシフトレジスタ11から出力させるビット列のビット数 $2ms$ は、線形フ

ィードバックシフトレジスタ 11 の 1 周期分のビット数 m ($= (2^n) - 1$) を 2 倍した値と、導出値 s とを乗算することによって求められる。

そして次に、線形フィードバックシフトレジスタ 11 から初期値をもとに $2ms$ 個のビット数を有するビット列を出力させ (ステップ S 4)、そのビット列から新ビット列を生成する (ステップ S 5)。新ビット列は、 $2ms$ 個のビット列から導出値 s の間隔ごとに取り出したビット列によって構成され、そのビット数は $2m$ 個となる。

ここで、出力系列が M 系列のビット列を s 個ごとにサンプルしたビット列は、その M 系列の 1 周期分のビット数 m と導出値 s とが互いに素であれば、他の構成を有する線形フィードバックシフトレジスタの M 系列となることから、この新ビット列も、M 系列となる。

そして、その新ビット列に基づいて線形フィードバックシフトレジスタ 11 の構成を再構成する (ステップ S 6)。線形フィードバックシフトレジスタ 11 の再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われる。バーレイキャンプマッセイアルゴリズムによれば、少なくとも 2 周期分以上のビット数を有するビット列があれば、かかるビット列を出力可能な等価で最小の線形フィードバックシフトレジスタを求めることができるので、 $2m$ 個のビット数を有する新ビット列から新たな線形フィードバックシフトレジスタの特性多項式を導出して、再構成を行う。

再構成後の線形フィードバックシフトレジスタ 11 は、再構成前と同一の次数及び異なる結線の特性多項式を有し、同一の初期値を与えた場合に、再構成前と異なる M 系列を出力可能な構成を有する。

線形フィードバックシフトレジスタ再構成手段 12 による線形フィードバックシフトレジスタ 11 の再構成が終了すると、再構成された線形フィードバックシフトレジスタ 11 から初期値をもとに疑似乱数を発生させる処理が行われる (ステップ S 7)。これにより、疑似乱数生成部 10 から再構成前とは異なる M 系列の疑似乱数が発生される。

各疑似乱数生成部 10 から出力された疑似乱数は、それぞれ非線形変換部 20 に入力され、非線形変換部 20 で所定の非線形関数 $f(x)$ に基づいて非線形変

換される（ステップS 8）。これにより、疑似乱数に非線形性を与えることができ、暗号強度を更に向上させることができる。

上記構成を有する疑似乱数発生器 1 によれば、線形フィードバックシフトレジスタ 11 の構成を初期値に基づいて容易かつ動的に変更することができ、変更後も M 系列を出力させることができる。したがって、解読者は、再構成前の線形フィードバックシフトレジスタの構成を取得することができない。これにより、従来、線形フィードバックシフトレジスタの構成が既知であることを前提に成り立っていた既存の暗号解読法は、成立しなくなる。したがって、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

尚、上述の実施の形態では、非線形コンバイナ型の疑似乱数発生器 1 を例に説明したが、非線形コンバイナ型に限定されるものではなく、線形フィードバックシフトレジスタを用いる疑似乱数発生器であればよく、例えばブロック型暗号方式に用いられる疑似乱数発生器に用いてもよい。

また、上記のステップ S 6 で、新ビット列に基づいて線形フィードバックシフトレジスタ 11 の構成を再構成する代わりに、新ビット列を出力可能な構成を有する第 2 の線形フィードバックシフトレジスタを生成し、ステップ S 7 で、その第 2 の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させてもよい。これによれば、線形フィードバックシフトレジスタを 2 つに分けることができ、より秘匿性の向上を図ることができる。また、第 1 の実施の形態における疑似乱数発生器 1 は、ソフトウェアやハードウェアのいずれによって構成してもよい。

（第 2 の実施の形態）

次に、本発明の第 2 の実施の形態について図に基づいて説明する。

第 4 図は、第 2 の実施の形態における疑似乱数発生器 1 の機能を概略的に示す説明図である。本実施の形態における疑似乱数発生器 1 は、疑似乱数発生プログラムをコンピュータハードウェア上で実行することによって実現される非線形コンバイナ型の疑似乱数発生器 1 である。尚、本実施の形態では、暗号化装置（従来技術を参照）に組み込まれた場合のものを例に説明し、復号化装置のものについては同様であるのでその詳細な説明を省略する。

疑似乱数発生器 1 は、第 4 図に示すように、乱数ビット列出力部 5 0 と、乱数ビット列増幅部 6 0 と、非線形変換部 8 0 を有している。乱数ビット列出力部 5 0 は、 α 個の選択用乱数ビット列出力手段 5 1 を備えている。選択用乱数ビット列出力手段 5 1₁ ~ 5 1 _{α} は、利用者から与えられる L_k ビットの秘密鍵 K をもと

5 に N_i ビットを有する選択用乱数ビット列を連続して出力するものであり、例えば線形フィードバックシフトレジスタによって構成される。

乱数ビット列増幅部 6 0 は、 N_i ビットの選択用乱数ビット列を与えることにより N_i ビットよりも大きなビット数である N_o ビットの増幅乱数ビット列を出力するように構成されており、乱数テーブル部 6 1 と排他的論理和演算処理手段

10 6 3 を備えている。

乱数テーブル部 6 1 は、 (2^{N_i}) 個の乱数ビット列を格納した $\alpha \times \beta$ (以下、単に「 $\alpha \beta$ 」と記す) 個の乱数テーブル 6 2 によって構成されている。そして、第 4 図に示すように、各選択用乱数ビット列出力手段 5 1 ごとに β (複数) 個の乱数テーブル 6 2 が対応するように設けられている。第 5 図は、一の乱数テ

15 ーブルの構成を説明する概略図である。各乱数テーブル 6 2 は、第 5 図に示すように、 (2^{N_i}) 個でかつ $0 \sim (2^{N_i}) - 1$ のインデックス番号が付与されたインデックス部 R_i と、各インデックス番号に一对一で対応して設けられ上述の増幅乱数ビット列を格納可能なビット列格納部 R_o を有している。

そして、乱数ビット列出力部 5 0 の選択用乱数ビット列出力手段 5 1 から出力

20 された選択用乱数ビット列を引数として、該当するインデックス部 R_i のインデックス番号を選択し、そのインデックス番号に対応する乱数ビット列格納部 R_o から N_o ビットの増幅乱数ビット列を選択できるように構成されている。

排他的論理和演算処理手段 6 3 は、乱数テーブル 6 2₁ ~ 6 2 _{$\alpha \beta$} の参照によって抽出された $\alpha \beta$ 個の増幅乱数ビット列を各選択用乱数ビット列出力手段 5 1 ごとに排他的論理和演算処理し、 α 個の増幅乱数ビット列とし、非線形変換部 8 0 に出力するように構成されている。これにより、乱数テーブル 6 2₁ ~ 6 2 _{$\alpha \beta$} から読み出した増幅乱数ビット列をそのまま非線形変換部 8 0 に出力するのではなく、暗号強度が増幅乱数ビット列そのものに依存することを防止し、暗号強度を更に向上させている。

第6図は、乱数ビット列増幅部60内に構成される各構成要素を説明する概念図である。上述の乱数ビット列増幅部60は、第6図に示すように、その内部機構として増幅乱数ビット列選択手段64を備えている。増幅乱数ビット列選択手段64は、各選択用乱数ビット列出力手段51₁～51_αから出力された選択用乱数ビット列を引数として各乱数テーブル62₁～62_{αβ}をそれぞれ参照し、引数と等しい値を有するインデックス番号に対応するビット列格納部R₀から増幅乱数ビット列をそれぞれ選択するように構成されている。

また、乱数ビット列増幅部60は、乱数テーブル部61の初期設定を行う乱数テーブル初期設定手段65と、その乱数テーブル初期設定手段65により乱数テーブル部61内に設定される増幅乱数ビット列を発生させる増幅乱数ビット列発生手段66を備えている。

乱数テーブル初期設定手段65は、増幅乱数ビット列発生手段66によって発生させた乱数ビット列をN₀ビットごとに分割して各乱数テーブル62₁～62_{αβ}の全ての乱数ビット列格納部R₀に格納する処理を行うものであり、本実施の形態では、第1番目の選択用乱数ビット列出力手段51₁に対応する乱数テーブル62₁から第α番目の選択用乱数ビット列出力手段51_αに対応する乱数テーブル62_{αβ}まで順番に格納するように構成されている。

増幅乱数ビット列発生手段66は、秘密鍵Kをもとに乱数ビット列を出力するものであり、本実施の形態では、RC4 Sympetric Stream Cipher (RSA Data Security Inc. 製) を用いている。しかし、通常の線形フィードバックシフトレジスタなどの疑似乱数ビット列を高速に出力できるもの（主としてストリーム型暗号）であれば他のものであってもよい。

また、第6図に示すように、乱数ビット列増幅部60は、所定のタイミングで乱数テーブル62₁～62_{αβ}の順番を入れ替える処理を行う乱数テーブル入れ替え手段67と、その乱数テーブル入れ替え手段67が乱数テーブルの順番入れ替え処理を行うために使用する順番入れ替え用乱数を発生させる入れ替え用乱数発生手段68を備えている。

乱数テーブル入れ替え手段67は、入れ替え用乱数発生手段68により発生さ

せた入れ替え用乱数を、乱数テーブルのテーブル番号として、その発生させた順番で各乱数テーブル $62_1 \sim 62_{\alpha\beta}$ に順次付与し、その付与した乱数をもとに乱数テーブルの順番を入れ替える処理を行い、乱数テーブル部 61 内の増幅乱数ビット列の順番をテーブル単位で変更する。

- 5 入れ替え用乱数発生手段 68 は、任意の秘密鍵 K_0 をもとに乱数テーブル入れ替え用乱数を発生させる処理を行うものであり、乱数ビット列出力部 50 から N_i ビットを有する α 個の選択用乱数ビット列を入力するごとに、 $\alpha\beta$ 個の入れ替え用乱数を発生するように構成されている。任意の秘密鍵 K_0 は、本実施の形態では、上述の増幅乱数ビット列発生手段 66 に秘密鍵 K を与えて出力させた増幅
- 10 乱数ビット列から L_k ビット分だけ取り出した値を用いている。しかし、これに拘束されるものではなく、例えば、他の手段によって発生させたり、別途にユーザに入力させてもよい。

- 非線形変換部 80 は、 α 入力 1 出力の 1 次無相関な非線形関数 $f(x)$ を有しており、乱数ビット列増幅部 60 から出力された α 個の増幅乱数ビット列を非線
- 15 形変換し、 N_o ビットを有する 1 個の乱数ビット列を疑似乱数 Z として出力するように構成されている。

- 尚、秘密鍵 K は、128 ビット、256 ビット、512 ビット、1024 ビットの中から選択され、また、選択用乱数ビット列出力手段 51 の数 α 、各選択用乱数ビット列出力手段 51 に対応する乱数テーブルの数 β 、及び選択用乱数ビッ
- 20 ト列のビット数 N_i は、互いにかけ算した値が秘密鍵 K のビット数 L_k に等しいという条件の範囲内で選択される。

次に、疑似乱数発生方法について第 7 図に基づき説明する。第 7 図は、本実施の形態における疑似乱数発生方法を説明するフローチャートである。

- まず最初に、乱数ビット列出力部 50 は、ユーザから L_k ビットを有する任意
- 25 の秘密鍵 K の入力を受けると（ステップ S11）、その秘密鍵 K を用いて選択用乱数ビット列出力手段 51 の初期値を設定する（ステップ S12）。例えば、選択用乱数ビット列出力手段 51 が線形フィードバックシフトレジスタによって構成されている場合には、その秘密鍵 K に基づいて各シフトレジスタ内に格納される初期値の設定が行われる。

各選択用乱数ビット列出力手段51の初期値を設定すると、次に、乱数テーブル初期設定手段65により乱数テーブル部61の初期設定が行われる（ステップS13）。ここでは、まず、増幅乱数ビット列発生手段66に秘密鍵Kが与えられ、高速で乱数ビット列が発生される。この増幅乱数ビット列発生手段24により発生された乱数ビット列は、乱数テーブル初期設定手段65によって、 N_i ビットごとに分割され、増幅乱数ビット列として各乱数テーブル62₁～62 _{$\alpha\beta$} の全ての乱数ビット列格納部R₀に順次格納される。このように、秘密鍵Kが与えられることによって、乱数テーブル部61の初期設定が予め行われる。

10 上述のステップS11～ステップS13により選択用乱数ビット列出力手段51と乱数テーブル部61の初期値の設定が行われると、待機状態となる。そして、平文の暗号化装置（従来技術を参照）への入力をトリガとして、乱数ビット列の増幅処理が開始される（ステップS14～S16）。まず最初に、各選択用乱数ビット列出力手段51によって、それぞれ N_i ビットを有する選択用乱数ビット列を乱数テーブルの数である β 個出力させ、乱数ビット列増幅部60内に記憶させる（ステップS14）。

15 それから、乱数テーブル順番入れ替え手段26により乱数テーブル62₁～62 _{$\alpha\beta$} の順番入れ替え処理を行う（ステップS15）。ここでは、まず、入れ替え用乱数発生手段68により $\alpha\beta$ 個の入れ替え用乱数を発生させ、乱数テーブルの順番入れ替え用のテーブル番号として、各乱数テーブル62₁～62 _{$\alpha\beta$} に付与する。20 これらのテーブル番号は、その発生の順番で乱数テーブル62₁から順次付与される。

したがって、各乱数テーブル62₁～62 _{$\alpha\beta$} には、1～ $\alpha\beta$ までのテーブル番号が順不同に付与される。そして、その付与したテーブル番号をもとに乱数テーブル部61内の増幅乱数ビット列の順番を各乱数テーブル単位で入れ替える処理25が行われる。これにより、乱数テーブル部61の乱数ビット列格納部R₀に格納されている増幅乱数ビット列は、昇順や降順などの予め設定した規則に従って、各乱数テーブル単位で入れ替えられる。

乱数テーブル62₁～62 _{$\alpha\beta$} の順番入れ替える処理が終了すると、増幅乱数ビット列選択手段64により、各乱数テーブル62₁～62 _{$\alpha\beta$} 内から該当する増幅

乱数ビット列を選択する増幅乱数ビット列選択処理が行われる(ステップS16)。増幅乱数ビット列選択手段64は、乱数ビット増幅部20に格納されている各選択用乱数ビット列を用いて、対応する乱数テーブル62₁~62_{αβ}をそれぞれ参照し、各乱数テーブル62₁~62_{αβ}内からそれぞれ該当する増幅乱数ビット列

5 を選択する。

増幅乱数ビット列の選択処理が終了すると、次に、排他的論理和演算処理手段63により排他的論理和演算処理が行われる(ステップS17)。排他的論理和演算処理手段63は、各乱数テーブル62₁~62_{αβ}から読み出したαβ個の増幅乱数ビット列を、各選択用乱数ビット列出力手段51単位で排他的論理和演算

10 処理する。これにより、N₀ビットを有するα個の新たな増幅乱数ビット列が生成される。

そして、これらの新たに生成された増幅乱数ビット列は、非線形変換部80に出力され、非線形変換が行われる(ステップS18)。非線形変換部80は、予め設定された非線形関数に基づいてN₀ビットのαβ個の増幅乱数ビット列を非線形変換し、N₀ビットを有する1個の乱数ビット列を疑似乱数として出力する。

15

非線形変換部80から疑似乱数を出力すると、再びステップS14まで戻り、ステップS14からステップS18までの処理を繰り返し行う。そして、平文から暗号文に変換するために必要な分の疑似乱数を発生させる。

上述の疑似乱数発生器1によると、選択用乱数ビット列出力手段51によって出力したN_iビットの選択用乱数ビット列に基づき乱数テーブルを参照することで、N_iビットよりも大きなビット数を有するN₀ビットの増幅乱数ビット列を非線形変換部80に供給することができる。したがって、従来はボトルネックとなっていた非線形変換部80よりも上流側のスループットを向上させることができ、非線形変換部80のスループットに近づけることができる。したがって、疑似乱数発生器1全体のスループットを高速化することができる。

20

25

また、選択用乱数ビット列出力部20からの選択用乱数ビット列の入力に応じて、乱数テーブル順番入れ替え処理を行うので、疑似乱数の暗号強度を増大させることができる。特に、本実施の形態では、乱数テーブル62₁~62_{αβ}の組合せパターンを(αβ)の階乗個(以下、階乗を「!」で表す)にすることができ

る。したがって、乱数テーブル部 61 の内容を既知と仮定したときに成立する攻撃では、 $(2^{(\alpha \beta \times N_i)}) \times (\alpha \beta) !$ の計算量が必要となり、この計算量は、 L_k ビットの秘密鍵を全数探索する場合の計算量よりも多くなることから、十分な暗号強度を備えていることがわかる。

- 5 また、上述の疑似乱数発生器 1 は、一の選択用乱数ビット列出力手段 51 から出力した乱数ビット列を用いて複数 (β 個) の乱数テーブルを参照し、各乱数テーブルから選択した乱数ビット列に排他的論理和演算を施す処理を行っている。したがって、乱数テーブル 61 から読み出した増幅乱数ビット列をそのまま非線形変換部 80 に出力した場合のように暗号強度が増幅乱数ビット列発生手段 66
- 10 そのものに依存するのを防ぎ、暗号強度を更に向上させている。

次に、本実施の形態における具体的な一実施例について説明する。第 8 図は、本実施例の疑似乱数発生器 1 を概略的に示す概念図、第 9 図は、乱数テーブル部 61 を概略的に示す概念図である。尚、本実施例では、各設定値 (パラメータ) を以下のように設定した場合を例に説明する。

- 15 選択用乱数ビット列出力手段の数: 8 個 ($\alpha = 8$)
 各選択用乱数ビット列出力手段に対応した乱数テーブルの数: 2 個 ($\beta = 2$)
 乱数テーブルのインデックス部の長さ: 2^8 個 ($N_i = 8$)
 乱数テーブルの乱数ビット列部の長さ: 2^{16} 個 ($N_o = 16$)
 秘密鍵の長さ: 128 ビット ($L_k = 128$)
- 20 非線形変換部 80 の非線形関数 $f(x)$:

$$\begin{aligned}
 f(x) = & x_1 + x_5 \\
 & + x_1x_2 + x_1x_3 + x_2x_3 + x_2x_5 + x_2x_6 + x_3x_6 \\
 & + x_1x_7 + x_2x_7 + x_4x_8 + x_5x_8 \\
 & + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 + x_1x_2x_5 \\
 25 & + x_2x_4x_5 + x_3x_4x_5 + x_1x_2x_6 + x_2x_3x_6 + x_1x_4x_6 \\
 & + x_4x_5x_6 + x_1x_2x_7 + x_2x_3x_7 + x_1x_4x_7 + x_1x_5x_7 \\
 & + x_2x_5x_7 + x_4x_5x_7 + x_1x_6x_7 + x_4x_6x_7 + x_5x_6x_7 \\
 & + x_1x_2x_8 + x_1x_3x_8 + x_2x_3x_8 + x_3x_4x_8 + x_1x_5x_8 \\
 & + x_3x_5x_8 + x_4x_5x_8 + x_3x_6x_8 + x_4x_6x_8 + x_5x_6x_8
 \end{aligned}$$

$$\begin{aligned}
& + x_1x_7x_8 + x_2x_7x_8 \\
& + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_4x_6 \\
& + x_1x_3x_4x_6 + x_2x_3x_4x_6 + x_1x_4x_5x_6 + x_2x_4x_5x_6 \\
& + x_3x_4x_5x_6 + x_1x_2x_3x_7 + x_1x_2x_4x_7 + x_2x_3x_4x_7 \\
5 \quad & + x_1x_2x_5x_7 + x_1x_4x_5x_7 + x_2x_4x_5x_7 + x_1x_2x_6x_7 \\
& + x_1x_3x_6x_7 + x_2x_3x_6x_7 + x_1x_4x_6x_7 + x_2x_4x_6x_7 \\
& + x_3x_4x_6x_7 + x_1x_5x_6x_7 + x_2x_5x_6x_7 + x_3x_5x_6x_7 \\
& + x_1x_2x_4x_8 + x_1x_2x_5x_8 + x_1x_3x_5x_8 + x_1x_4x_5x_8 \\
& + x_1x_2x_6x_8 + x_2x_3x_6x_8 + x_1x_4x_6x_8 + x_2x_5x_6x_8 \\
10 \quad & + x_3x_5x_6x_8 + x_1x_3x_7x_8 + x_1x_4x_7x_8 + x_2x_4x_7x_8 \\
& + x_3x_4x_7x_8 + x_2x_5x_7x_8 \\
& + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6 + x_1x_3x_4x_5x_6 \\
& + x_2x_3x_4x_5x_6 + x_1x_2x_4x_5x_7 + x_2x_3x_4x_5x_7 \\
& + x_1x_2x_4x_6x_7 + x_1x_3x_4x_6x_7 + x_1x_4x_5x_6x_7 \\
15 \quad & + x_2x_4x_5x_6x_7 + x_1x_2x_3x_4x_8 + x_1x_2x_3x_5x_8 \\
& + x_1x_2x_4x_5x_8 + x_1x_2x_3x_6x_8 + x_1x_2x_4x_6x_8 \\
& + x_1x_3x_4x_6x_8 + x_2x_3x_5x_6x_8 + x_1x_4x_5x_6x_8 \\
& + x_2x_4x_5x_6x_8 + x_1x_2x_3x_7x_8 + x_1x_3x_4x_7x_8 \\
& + x_1x_3x_5x_7x_8 + x_2x_3x_5x_7x_8 + x_3x_4x_5x_7x_8 \\
20 \quad & + x_1x_3x_6x_7x_8 + x_3x_4x_6x_7x_8 \\
& + x_1x_2x_3x_4x_5x_8 + x_1x_2x_3x_4x_6x_8 \\
& + x_1x_3x_4x_5x_6x_8 + x_2x_3x_4x_5x_6x_8 \\
& + x_1x_2x_3x_4x_7x_8 + x_1x_2x_3x_5x_7x_8 \\
& + x_1x_2x_4x_5x_7x_8 + x_1x_3x_4x_5x_7x_8 \\
25 \quad & + x_1x_3x_4x_6x_7x_8 + x_2x_3x_4x_6x_7x_8 \\
& + x_1x_2x_5x_6x_7x_8 + x_1x_3x_5x_6x_7x_8
\end{aligned}$$

本実施例では、各選択用乱数ビット列出力手段 5 1 が、ユーザから与えられる秘密鍵 K に基づいて線形フィードバックシフトレジスタ 5 3 の再構成を行い、その再構成後の線形フィードバックシフトレジスタ 5 3' を用いて選択用乱数ビッ

ト列を出力するように構成されている。

まず最初に、この選択用乱数ビット列出力手段 5 1 の構成及びその動作について説明する。選択用乱数ビット列出力手段 5 1 は、第 8 図に示すように、初期値設定手段 1 2、線形フィードバックシフトレジスタ 5 3、線形フィードバックシフトレジスタ再構成手段 1 4 を有している。

初期値設定手段 1 2 は、ユーザから与えられる秘密鍵 K に基づいて初期値を設定するものであり、秘密鍵 K をビット列に変換し、初期値として線形フィードバックシフトレジスタ 5 3 のシフトレジスタ内に割り当てるように構成されている。初期値設定手段 1 2 は、本実施例では、RC4 S y p p e t r i c S t r e a p C i p h e r (RSA Data Security Inc. 製) を用いており、増幅乱数ビット列発生手段 6 6 と共用している。

線形フィードバックシフトレジスタ 5 3 は、従来技術で説明したものと同様に、1 ビットの情報を記憶できる n 個のシフトレジスタと、排他的論理和演算回路を有している。そして、本実施の形態では、1 周期分のビット数 m が $(2^n) - 1$ 個となるビット列、いわゆる M 系列を出力可能な構成に予め設定されている。

第 1 1 図は、本実施の形態における線形フィードバックシフトレジスタ 5 3 の初期多項式を例示するものである。初期多項式は、M 系列を出力するように予め設定されている特性多項式であり、1 項目の指数部分がシフトレジスタの個数を示し、2 項目以降の指数部分が排他的論理和演算回路に接続された結線位置を示している。例えば、1 段目の線形フィードバックシフトレジスタ (LFSR 1) 5 3 は、1 2 9 個のシフトレジスタを有し、8 0 番目、8 番目、1 番目のシフトレジスタがフィードバックタップによって排他的論理和演算回路に接続されていることを示している。尚、本実施の形態では、シフトレジスタの個数 n は、全て素数個に設定されている。

線形フィードバックシフトレジスタ再構成手段 1 4 は、線形フィードバックシフトレジスタ 5 3 の構成を秘密鍵 K によって動的に変更して再構成するものである。具体的には、出力系列が M 系列のビット列を s 個ごとにサンプルした新ビット列は、M 系列の 1 周期分のビット数 $m (= (2^n) - 1)$ と導出値 s とが互いに素であるとき、すなわち、1 以外の共通の約数を持たないときは、他の構成

を有する線形フィードバックシフトレジスタのM系列になり、また、バーレイキャン
プマッセイアルゴリズムによって、少なくとも2周期分以上のビット数を有
するビット列から、そのビット列を出力可能な等価で最小の線形フィードバック
シフトレジスタの特性多項式を求めることができることを利用して、線形フィード
バックシフトレジスタ53の再構成を行う。

線形フィードバックシフトレジスタ再構成手段14は、初期値設定手段12に
よって与えられた初期値から導出値sを算出し、導出値sと線形フィードバック
シフトレジスタ53の1周期分のビット数 $m (= (2^n) - 1)$ を2倍した値
2mとを乗算し、線形フィードバックシフトレジスタ53から出力させるビット
列のビット数2msを算出する。

そして、初期値をもとに線形フィードバックシフトレジスタ53から2ms個
のビット列を出力させ、その2ms個のビット列から導出値sの間隔ごとにビッ
ト列を取り出して新ビット列を生成し、その新ビット列を用いてバーレイキャン
プマッセイアルゴリズムにより線形フィードバックシフトレジスタ53の構成を
変更する。

尚、線形フィードバックシフトレジスタ53から出力させるビット列のビット
数は、新ビット列のビット数が2m個以上であれば、等価な最小の線形フィード
バックシフトレジスタを求めることができるので、2ms個以上であればよい。

バーレイキャンプマッセイアルゴリズムとは、線形フィードバックシフトレジ
スタ53のシフトレジスタの個数n（線形複雑度）の2倍以上のビット数を有す
るビット列を入手することで、そのビット列を出力可能な等価な最小の線形フィ
ードバックシフトレジスタを得ることができるというアルゴリズムである。バー
レイキャンプマッセイアルゴリズムについては、例えば、文献1「暗号理論入門
（第2版）」、共立出版社、岡本栄司著、2002年4月10日発行、に詳細に
説明されている。

第12図は、線形フィードバックシフトレジスタ53の再構成処理を説明する
フローチャートである。まず最初に、初期値設定手段12によって初期値が設定
される（ステップS41）。初期値は、利用者から与えられるLkビットの秘密
鍵Kに基づいて設定される。初期値設定手段12において秘密鍵Kから初期値が

設定されると、その初期値は、線形フィードバックシフトレジスタ 53 のシフトレジスタ内にセットされる。

次に、所定の演算処理により初期値から線形フィードバックシフトレジスタ 53 の 1 周期分のビット数 m と互いに素である導出値 s を算出する (ステップ S 4 2) 。導出値 s は、初期値に対して、例えば MD 5 (Message Digest 5) などのハッシュ関数を施してハッシュ値を求め、そのハッシュ値に最も近似した素数が採用される。導出値 s は、初期値から求めることができ、かつビット数 m と互いに素であればよく、上記の算出方法によって求められるものに限定されない。但し、秘匿性を維持するために、上記所定の演算処理は、一方向性を満足しうる演算処理でなければならない。

導出値 s を算出すると、次に、線形フィードバックシフトレジスタ 53 から出力させるビット列のビット数 $2ms$ を算出する (ステップ S 4 3) 。線形フィードバックシフトレジスタ 53 から出力させるビット列のビット数 $2ms$ は、線形フィードバックシフトレジスタ 53 の 1 周期分のビット数 m ($= (2^n) - 1$) を 2 倍した値と、導出値 s とを乗算することによって求められる。

そして次に、線形フィードバックシフトレジスタ 53 から初期値をもとに $2ms$ 個のビット数を有するビット列を出力させ (ステップ S 4 4) 、そのビット列から新ビット列を生成する (ステップ S 4 5) 。新ビット列は、 $2ms$ 個のビット列から導出値 s の間隔ごとに取り出したビット列によって構成され、そのビット数は $2m$ 個となる。

ここで、出力系列が M 系列のビット列を s 個ごとにサンプルしたビット列は、その M 系列の 1 周期分のビット数 m と導出値 s とが互いに素であれば、他の構成を有する線形フィードバックシフトレジスタの M 系列となることから、この新ビット列も、M 系列となる。

そして、その新ビット列に基づいて線形フィードバックシフトレジスタ 53 の構成を再構成する (ステップ S 4 6) 。線形フィードバックシフトレジスタ 53 の再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われる。バーレイキャンプマッセイアルゴリズムによれば、少なくとも 2 周期分以上のビット数を有するビット列があれば、かかるビット列を出力可能な等価で最小の線形フィ

ードバックシフトレジスタ 53 を求めることができるので、 2^m 個のビット数を有する新ビット列から新たな線形フィードバックシフトレジスタ 53 の特性多項式を導出して、再構成を行う。

- 5 再構成後の線形フィードバックシフトレジスタ 53' は、再構成前と同一の次数及び異なる結線の特性多項式を有し、同一の初期値を与えた場合に、再構成前と異なるM系列を出力可能な構成を有する。

- 10 線形フィードバックシフトレジスタ再構成手段 14 による線形フィードバックシフトレジスタ 53 の再構成が終了すると、再構成された線形フィードバックシフトレジスタ 53' から初期値をもとに乱数ビット列を発生させる処理が行われる (ステップ S47)。これにより、乱数ビット列出力部 50 からは再構成前とは異なるM系列の乱数ビット列が出力される。

- 15 尚、上記のステップ S46 で、新ビット列に基づいて線形フィードバックシフトレジスタ 53 の構成を再構成する代わりに、新ビット列を出力可能な構成を有する第2の線形フィードバックシフトレジスタを生成し、ステップ S47 で、その第2の線形フィードバックシフトレジスタによって初期値をもとに乱数ビット列を発生させてもよい。これによれば、線形フィードバックシフトレジスタ 53 を2つに分けることができ、より秘匿性の向上を図ることができる。

- 20 上記の選択用乱数ビット列出力手段 51 は、線形フィードバックシフトレジスタ 53 の構成を初期値に基づいて容易かつ動的に変更することができ、変更後もM系列を出力させることができる。したがって、攻撃者は、再構成前の線形フィードバックシフトレジスタ 53 の構成を取得することができない。これにより、従来、線形フィードバックシフトレジスタ 53 の構成が既知であることを前提に成り立っていた既存の暗号解読法は、成立しなくなる。したがって、高い暗号強度を得ることができ、情報の秘匿性を保つことができる。

- 25 次に、上記の選択用乱数ビット列出力手段 51 を備えた疑似乱数発生器 1 による疑似乱数発生方法について説明する。第10図は、本実施例における疑似乱数発生方法を説明するフローチャートである。

まず最初に、乱数ビット列出力部 50 は、ユーザから 128 ビット ($L_k = 128$) を有する任意の秘密鍵 K の入力を受け取ると、その秘密鍵 K に基づいて再

構成前の線形フィードバックシフトレジスタ 5 3 の初期値を設定する（ステップ S 2 1）。

- そして、その初期値に基づいて線形フィードバックシフトレジスタ 5 3 を再構成し（ステップ S 2 2）、再構成後の線形フィードバックシフトレジスタ 5 3' に初期値を設定する（ステップ S 2 3）。この初期値の設定を、全ての乱数ビット列出力手段 $1\ 1_1 \sim 1\ 1_8$ について行う。

- 次に、乱数ビット列増幅部 6 0 は、乱数テーブル部 6 1 の初期設定を行う（ステップ S 2 4）。ここでは、まず、増幅乱数ビット列発生手段 6 6 に秘密鍵 K を与え、高速で乱数ビット列を発生させる処理が行われるが、本実施例では、上述のように、増幅乱数ビット列発生手段 6 6 と選択用乱数ビット列出力手段 5 1 の初期値設定手段 1 2 とを共用しているので、別途出力することはせずに、線形フィードバックシフトレジスタ 5 3 の初期値として出力した乱数ビット列をそのまま用いる。

- 乱数テーブル初期設定手段 6 5 は、その乱数ビット列を 1 6 ビット ($N_o = 16$) ごとに分割し、増幅乱数ビット列として各乱数テーブル $6\ 2_1 \sim 6\ 2_{16}$ の全ての乱数ビット列格納部 R_o に順次格納する。

以上の初期値設定段階が終了すると（ステップ S 2 1 ~ S 2 4）、待機状態となる。そして、平文の暗号化装置（従来技術を参照）への入力をトリガとして、疑似乱数を発生させる処理（ステップ S 2 5 ~ S 2 7）に移行する。

- ここでは、各選択用乱数ビット列出力手段 $5\ 1_1 \sim 5\ 1_8$ ごとにそれぞれ選択用乱数ビット列を出力させ、乱数ビット列増幅部 6 0 のバッファ内にそれぞれ記憶させる処理が行われる。具体的には、各選択用乱数ビット列出力手段 $5\ 1_1 \sim 5\ 1_8$ から 8 ビットの選択用乱数ビット列がそれぞれ出力され（ステップ S 2 7）、その数が各選択用乱数ビット列出力手段 1 に対して 2 個分 ($\beta = 2$) であり（ステップ S 2 6 で Yes）、各選択用乱数ビット列出力手段 $5\ 1_1 \sim 5\ 1_8$ にそれぞれ対応する分である場合（ステップ S 2 5 で Yes）には、必要数の選択乱数ビット列が得られたとして次の乱数ビット列増幅段階に移行する。したがって、ここまでの処理により、バッファ内には 8 ビットを有する 1 6 個の選択用乱数ビット列が記憶される。

次に、秘密鍵K 0に基づき入れ替え用乱数発生手段6 8により1 6個の入れ替え用乱数を発生させ（ステップS 2 8）、乱数テーブルの順番入れ替え処理が行われる（ステップS 2 9）。ここでは、1 6個の乱数が順番入れ替え用のテーブル番号として、乱数テーブル6 2₁～6 2₁₆に付与される。したがって、1番～1 6番までのテーブル番号が順不同で乱数テーブル6 2₁～6 2₁₆に付与される。そして、その付与されたテーブル番号をもとに各乱数テーブル6 2₁～6 2₁₆の順番の入れ替えを行う。ここでは、選択用乱数ビット列出力手段5 1₁～5 1₁₆に対してテーブル番号が1番～1 6番に順番に並ぶように降順に入れ替える処理が行われる。これにより、乱数テーブル部6 1内の増幅乱数ビット列は、その順番が乱数テーブル単位でランダムに入れ替えられる。

そして次に、各乱数テーブル6 2₁～6 2₁₆内から該当する増幅乱数ビット列を選択する処理が行われる（ステップS 3 0～S 3 2）。例えば、選択用乱数ビット列1 1₁から出力されバッファに記憶された1番目の選択用乱数ビット列を引数として乱数テーブル6 2₁が参照される（ステップS 3 2）。そして、その引数と等しい値を有するインデックス番号を選択し、そのインデックス番号に対応する乱数ビット列格納部R oに格納された増幅乱数ビット列を選択する。

例えば、選択用乱数ビット列出力手段5 1₁から出力され乱数テーブル6 2₁に対応するものとしてバッファに記憶された選択用乱数ビット列が「0 0 0 0 0 0 1 1」である場合、これを8桁の2進数とみなし、1 0進数に変換して引数「3」を得る。この引数「3」を用いて乱数テーブル6 2₁を参照し、インデックス部R oのインデックス番号が「3」の乱数ビット列格納部R oに格納されている増幅乱数ビット列「0 1 0 1 1 0 1 0 1 1 0 1 1 1 0 1 1 0」を選択する。

そして、乱数テーブル6 2₁と乱数テーブル6 2₂からそれぞれ増幅乱数ビット列を選択すると（ステップS 3 1でY e s）、これら2つの増幅乱数ビット列の排他的論理和演算処理を行い（ステップS 3 3）、1 6ビットを有する1個の新たな増幅乱数ビット列を生成する。

そして、同様の処理を各乱数テーブル6 2₃～6 2₁₆について行い（ステップS 3 0でY e s）、合計で8個の新たな増幅乱数ビット列を生成すると、非線形変換部8 0に出力して、非線形変換段階に移行する。

非線形変換部 80 では、乱数ビット列増幅部 60 よりこれらの N_0 ビットを有する 8 個の増幅乱数ビット列を入力すると、非線形関数 $f(x)$ により非線形変換し（ステップ S34）、16 ビットを有する 1 個の乱数ビット列を得る。そして、上記ステップ S25～ステップ S34 の処理を繰り返し実行して必要数の疑似乱数を得る。

本実施例については、処理速度の高速化及び乱数性が適切に確保されているかについて実験を行っており、その結果、従来と比較して 170 倍も処理速度を向上でき、同時に適切な乱数性も確保されているとの結果を得た。以下に、その実験内容及び実験結果について説明する。

- 10 実験に使用したコンピュータは、CPU: Pentium (登録商標) 4 (1.7 GHz)、メモリ: 256 MB である。また、各設定値は、上述の実施例と同一のものとする。そして、入れ替え用乱数ビット列発生手段 28 に用いられる秘密鍵 K_0 は、16 進数表記で以下のものに固定した状態として実験を行った。

$$K_0 = (f1e2d3c4b5a69788796a5b4c3d2e1f10)_{16}$$

- 15 第 13 図は、スループットの計測結果を示す表である。表中の従来型とは、8 個の線形フィードバックシフトレジスタ 53 と、非線形変換部 80 を用いて構成した、第 17 図に示すような従来の非線形コンパイナ型疑似乱数発生器を示す。

- 本実験結果によれば、第 13 図に示すように、疑似乱数発生器 1 の平均スループットが、線形フィードバックシフトレジスタ 53 そのものの平均スループットから、非線形変換部 80 の平均スループットに向上しており、更に、従来型の約 170 倍 ($116.4 \text{ Mbps/sec} \div 0.680 \text{ Mbps/sec} = 171.176 \dots$) になっていることがわかる。したがって、このスループット計測結果から、乱数テーブル 62 を用いたことが疑似乱数発生器 1 の高速化に有効であることがわかる。

本実施例における疑似乱数発生器 1 のスループットは、次式 (1) で表される。

- 25 【数式】

$$\frac{1}{T} = \frac{N_I}{N_0} \left(\frac{n}{T_1} + \frac{1}{T_2} + \frac{1}{T_3} \right) + \frac{nm}{T_4} + \frac{1}{T_5} \quad (1)$$

T_1 は、1 つの線形フィードバックシフトレジスタ 53 の平均スループットを示し、 T_2 は、RC4 (増幅乱数ビット列発生手段 66) の平均スループットを

示す。また、T 3 は、乱数テーブル入れ替え手段 6 7 による乱数テーブル入れ替え処理の平均スループットを示し、T 4 は、1 つの乱数テーブル 6 2 の平均スループットを示す。そして、T 5 は、非線形変換部 8 0 の平均スループットを示す。上記式 (1) から乱数テーブル 6 2 の計算量が無視できると仮定すると、 N_o ビット / N_i ビットを小さくするほど非線形変換部 8 0 のスループットに近づけることができ、高速化を図ることができる。

疑似乱数の暗号強度の検証については、NIST という疑似乱数検定ツールを用いて検定を行った。NIST とは、物理乱数及び疑似乱数生成器からの出力データについて乱数性のテストを行うツールであり、16 項目からのテストからなる統計のパッケージである。NIST については、<http://crsc.nist.gov/rug> に詳しく説明されている。第 14 図は、本検定に使用した NIST のパラメータを示す表である。各種テストを行うことによって出力された p -value が $0 < p\text{-value} < 1$ を満たす場合に、そのテスト項目をパスしたとみなしている。本実施例による疑似乱数発生器 1 の疑似乱数を検定したところ、全てのテスト項目をパスしていることが確認できた。第 15 図は、今回の実験による NIST の検定結果を示す図である。

尚、上述の実施例に示した各設定値は、暗号の安全性を確認するために設定したものであり、これに限定されるものではない。また、本発明は、上述の実施の形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で種々の変更、組み合わせが可能である。

[発明の効果]

以上説明したように、本発明に係る疑似乱数発生方法によれば、出力系列が M 系列のビット列を s 個ごとにサンプルしたビット列は、その M 系列の 1 周期分のビット数 $m (= (2^n) - 1)$ と導出値 s が互いに素であるときには、他の構成を有する線形フィードバックシフトレジスタの M 系列を構成し、また、バレーキャンパッセイアルゴリズムによって、少なくとも 2 周期分以上のビット数を有するビット列から線形フィードバックシフトレジスタを求めることができることを利用して、線形フィードバックシフトレジスタの構成を初期値に基づいて動的に変更することができ、変更後の線形フィードバックシフトレジスタから M

系列のビット列を出力させることができる。

したがって、解読者は、疑似乱数発生器から出力される疑似乱数に基づいて再構成前の線形フィードバックシフトレジスタの構成を得ることができず、初期値や秘密鍵も解読することができない。この結果、高い暗号強度を得ることができ、

5 情報の秘匿性を保つことができる。

また、他の発明によれば、秘密鍵に基づいて所定のビット数を有する選択用乱数ビット列を出力させ、その選択用乱数ビット列を用いて乱数テーブルを参照することにより、乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択し、非線形変換手段によって非線形変換して疑似乱数として

10 出力させるので、小さなビット列を有する選択用乱数ビット列に基づいて、より大きなビット数を有する増幅乱数ビット列を得ることができる。

したがって、非線形変換手段に入力される乱数ビット列をより大きなビット数を有するものにすることができる。これにより、従来、ボトルネックとなっていた非線形変換手段よりも上流側の乱数ビット列を出力する部分のスループットを

15 向上させ、非線形変換手段のスループットに近づけることができ、疑似乱数発生器全体のスループットを高速化することができる。

[図面の簡単な説明]

[第1図]

本実施の形態における疑似乱数発生器を説明する図である。

20 [第2図]

本実施の形態における線形フィードバックシフトレジスタの初期多項式を例示するものである。

[第3図]

本実施の形態における疑似乱数発生器の動作を説明するフローチャートである。

25 [第4図]

疑似乱数発生器の機能を概略的に示す説明図である。

[第5図]

乱数テーブル部の説明図である。

[第6図]

乱数ビット列増幅部内に構成される各構成要素を説明する概念図である。

〔第 7 図〕

本実施の形態における疑似乱数発生方法を説明するフローチャートである。

〔第 8 図〕

5 本実施例における疑似乱数発生器を概略的に示す概念図である。

〔第 9 図〕

乱数テーブル部を概略的に示す概念図である。

〔第 10 図〕

本実施例における疑似乱数発生方法を説明するフローチャートである。

10 〔第 11 図〕

本実施の形態における線形フィードバックシフトレジスタの初期多項式を例示するものである。

〔第 12 図〕

15 線形フィードバックシフトレジスタの再構成処理を説明するフローチャートである。

〔第 13 図〕

スループットの計測結果を示す表である。

〔第 14 図〕

本検定に使用した NIST のパラメータを示す表である。

20 〔第 15 図〕

NIST の検定結果を示す図である。

〔第 16 図〕

従来の逐次暗号方式を説明する図である。

〔第 17 図〕

25 暗号化装置の疑似乱数発生器を説明する図である。

〔第 18 図〕

一般的な線形フィードバックシフトレジスタの構成を簡単に説明する図である。

〔符号の説明〕

1 疑似乱数発生器

- 1 0 疑似乱数生成部
- 1 1 線形フィードバックシフトレジスタ
- 1 2 線形フィードバックシフトレジスタ再構成手段
- 2 0 非線形変換部
- 5 5 0 乱数ビット列出力部
- 5 1 選択用乱数ビット列出力手段
- 5 2 初期値設定手段
- 5 3 線形フィードバックシフトレジスタ
- 5 4 線形フィードバックシフトレジスタ再構成手段
- 10 6 0 乱数ビット列増幅部
- 6 1 乱数テーブル部
- 6 2₁ ~ 6 2_{αβ} 乱数テーブル
- 6 3 排他的論理和演算処理手段
- 6 4 増幅乱数ビット列選択手段
- 15 6 5 乱数テーブル初期設定手段
- 6 6 増幅乱数ビット列発生手段
- 6 7 乱数テーブル順番入れ替え手段
- 6 8 入れ替え用乱数発生手段
- 7 0 非線形変換部

請求の範囲

1. n 個のシフトレジスタを有し、1周期分のビット数が $(2^n) - 1$ 個となるビット列を出力可能な線形フィードバックシフトレジスタの初期値を設定
5 する第1ステップと、

所定の演算処理により前記初期値から前記線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める第2ステップと、

該導出値と前記線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値とを乗算して、前記第1線形フィードバックシフトレジスタにより
10 出力させるビット列のビット数を算出する第3ステップと、

前記算出したビット数分のビット列を前記線形フィードバックシフトレジスタから前記初期値をもとに出力させる第4ステップと、

該出力したビット列から前記導出値の間隔ごとにビット列を取り出して新ビット列を生成する第5ステップと、

15 該新ビット列を出力可能な構成に前記線形フィードバックシフトレジスタの構成を再構成する第6ステップと、

該再構成した後の線形フィードバックシフトレジスタから前記初期値をもとに疑似乱数を発生させる第7ステップと、

を有することを特徴とする疑似乱数発生方法。

20

2. 前記初期値に対してハッシュ関数を施してハッシュ値を求め、該ハッシュ値に最も近似した素数を導出値として採用することを特徴とする請求項1に記載の疑似乱数発生方法。

25 3. 前記線形フィードバックシフトレジスタの再構成は、バーレイキャンプマッセイアルゴリズムを用いて行われることを特徴とする請求項1または2に記載の疑似乱数発生方法。

4. 前記第7ステップで発生させた疑似乱数を非線形変換する第8ステップ

を有することを特徴とする請求項1～3のいずれかに記載の疑似乱数発生方法。

5. n 個のシフトレジスタを有し、1周期分のビット数が $(2^n) - 1$ 個となるビット列を出力可能な線形フィードバックシフトレジスタと、

5 秘密鍵に基づき前記線形フィードバックシフトレジスタの初期値を設定する初期値設定手段と、

所定の演算処理により前記初期値から前記線形フィードバックシフトレジスタの1周期分のビット数と互いに素である導出値を求める導出値算出手段と、

10 前記線形フィードバックシフトレジスタの1周期分のビット数を2倍以上した値と前記導出値とを乗算して、前記第1線形フィードバックシフトレジスタにより出力させるビット列のビット数を算出するビット数算出手段と、

前記算出したビット数分のビット列を前記線形フィードバックシフトレジスタから前記初期値をもとに出力させるビット列出力手段と、

15 該出力したビット列から前記導出値の間隔ごとにビット列を取り出して新ビット列を生成する新ビット列生成手段と、

該新ビット列を出力可能な構成に前記線形フィードバックシフトレジスタの構成を再構成する線形フィードバックシフトレジスタ再構成手段と、

20 該再構成後の線形フィードバックシフトレジスタから前記初期値をもとに疑似乱数を発生させる疑似乱数発生手段と、を有することを特徴とする疑似乱数発生器。

6. 前記線形フィードバックシフトレジスタ再構成手段の代わりに、新ビット列を出力可能な構成を有する第2の線形フィードバックシフトレジスタを生成する線形フィードバックシフトレジスタ生成手段を設け、

25 前記疑似乱数発生手段は、前記第2の線形フィードバックシフトレジスタによって初期値をもとに疑似乱数を発生させることを特徴とする請求項5に記載の疑似乱数発生器。

7. 秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力する

選択用乱数ビット列出力手段と、

前記選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、

前記選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて

- 5 前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、

該増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段と、

- 10 を有することを特徴とする疑似乱数発生器。

8. 前記秘密鍵が与えられることにより前記秘密鍵に基づいて前記増幅乱数ビット列を発生させ、前記乱数テーブルに格納して、前記乱数テーブルの初期設定を行う乱数テーブル初期設定手段を有することを特徴とする請求項7に記載の

15 疑似乱数発生器。

9. 前記選択用乱数ビット列出力手段は、複数設けられ、

前記乱数テーブルは、前記各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、

- 20 前記増幅乱数ビット列選択手段は、前記各選択用乱数ビット列出力手段から各々出力された前記各選択用乱数ビット列を用いて前記各選択用乱数ビット列出力手段ごとに対応する前記乱数テーブルをそれぞれ参照し、前記各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、

- 25 非線形変換手段は、前記各増幅乱数ビット列選択手段によって前記各乱数テーブルから選択された前記各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする請求項7または8に記載の疑似乱数発生器。

10. 前記各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブ

ルを設け、

前記増幅乱数ビット列選択手段により前記各乱数テーブル内から選択された各増幅乱数ビット列を前記選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段を有することを特徴とする

5 請求項 9 に記載の疑似乱数発生器。

1 1. 所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段を有することを特徴とする請求項 9 または 1 0 に記載の疑似乱数発生器。

10

1 2. 前記乱数テーブル入れ替え手段は、

前記各乱数テーブルを参照するために必要な選択用乱数ビット列を前記選択用乱数ビット列出力手段が出力するごとに、前記各乱数テーブル同士の入れ替えを行うことを特徴とする請求項 1 1 に記載の疑似乱数発生器。

15

1 3. 前記乱数テーブル入れ替え手段は、

前記各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、

20 該乱数テーブル入れ替え用乱数を前記乱数テーブルのテーブル番号として各乱数テーブルに付与し、前記テーブル番号をもとに予め設定された規則に従って前記乱数テーブルの順番を入れ替えることを特徴とする請求項 1 1 または 1 2 に記載の疑似乱数発生器。

1 4. コンピュータを、

25 秘密鍵に基づいて所定ビット数を有する選択用乱数ビット列を出力させる選択用乱数ビット列出力手段と、

前記選択用乱数ビット列よりも大きなビット数を有する増幅乱数ビット列を予め複数格納した乱数テーブルと、

前記選択用乱数ビット列出力手段から出力された選択用乱数ビット列を用いて

前記乱数テーブルを参照することにより、前記乱数テーブル内の複数の増幅乱数ビット列の中から該当する増幅乱数ビット列を選択可能な増幅乱数ビット列選択手段と、

- 5 該増幅乱数ビット列選択手段により選択された増幅乱数ビット列を非線形関数によって非線形変換し疑似乱数として出力する非線形変換手段として機能させるための疑似乱数発生プログラム。

- 10 15. 前記秘密鍵が与えられることにより前記秘密鍵に基づいて前記増幅乱数ビット列を発生させ、前記乱数テーブルに格納して、前記乱数テーブルの初期設定を行う乱数テーブル初期設定手段としてコンピュータを機能させることを特徴とする請求項14に記載の疑似乱数発生プログラム。

- 15 16. 前記選択用乱数ビット列出力手段は、複数設けられ、前記乱数テーブルは、前記各選択用乱数ビット列出力手段にそれぞれ対応するように設けられ、

前記増幅乱数ビット列選択手段は、前記各選択用乱数ビット列出力手段から各々出力された前記各選択用乱数ビット列を用いて前記各選択用乱数ビット列出力手段ごとに対応する前記乱数テーブルをそれぞれ参照し、前記各乱数テーブル内からそれぞれ該当する増幅乱数ビット列を選択し、

- 20 非線形変換手段は、前記各増幅乱数ビット列選択手段によって前記各乱数テーブルから選択された前記各増幅乱数ビット列を用いて非線形関数により非線形変換し疑似乱数として出力することを特徴とする請求項14または15に記載の疑似乱数発生プログラム。

- 25 17. 前記各選択用乱数ビット列出力手段ごとにそれぞれ複数の乱数テーブルを設け、

前記増幅乱数ビット列選択手段により前記各乱数テーブル内から選択された各増幅乱数ビット列を前記選択用乱数ビット列出力手段ごとに排他的論理和演算して非線形変換手段に出力する排他的論理和演算処理手段としてコンピュータを機

能させることを特徴とする請求項 16 に記載の疑似乱数発生プログラム。

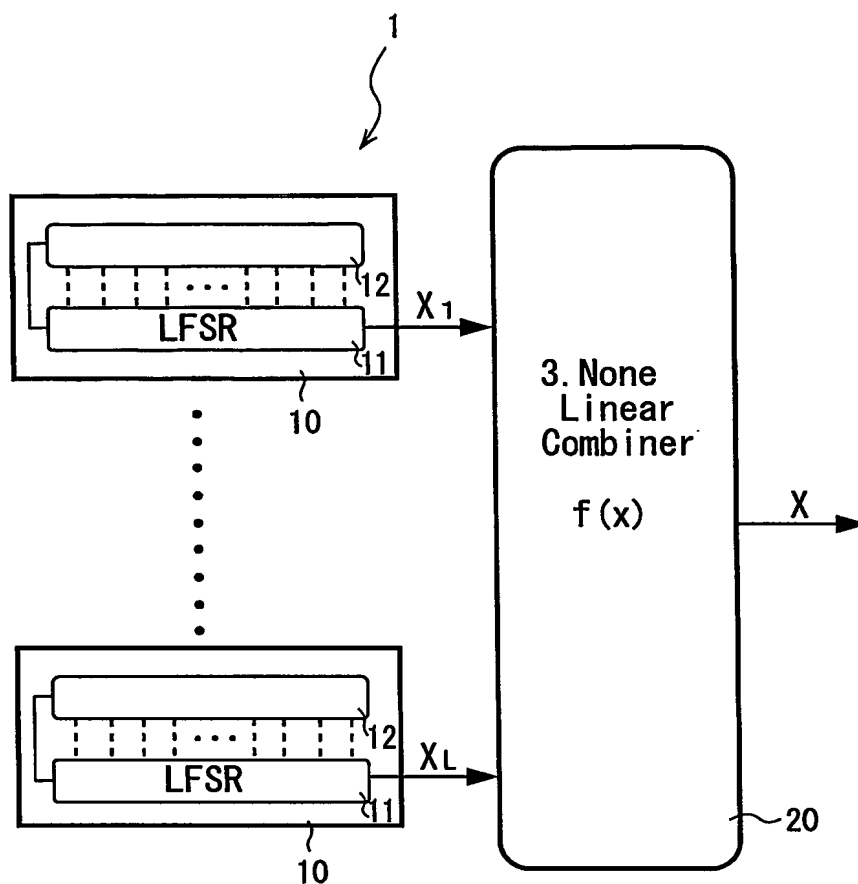
18. 所定のタイミングで前記各乱数テーブル同士の入れ替えを行う乱数テーブル入れ替え手段としてコンピュータを機能させることを特徴とする請求項 15 6 または 17 に記載の疑似乱数発生プログラム。

19. 前記乱数テーブル入れ替え手段は、
前記各乱数テーブルを参照するために必要な選択用乱数ビット列を前記選択用乱数ビット列出力手段が出力するごとに、前記各乱数テーブル同士の入れ替えを行うことを特徴とする請求項 18 に記載の疑似乱数発生プログラム。

20. 前記乱数テーブル入れ替え手段は、
前記各乱数テーブルの個数と等しい個数の乱数テーブル入れ替え用乱数を発生させ、
15 該乱数テーブル入れ替え用乱数を前記乱数テーブルのテーブル番号として各乱数テーブルに付与し、前記テーブル番号をもとに予め設定された規則に従って前記乱数テーブルの順番を入れ替えることを特徴とする請求項 18 または 19 に記載の疑似乱数発生プログラム。

1/15

第1図

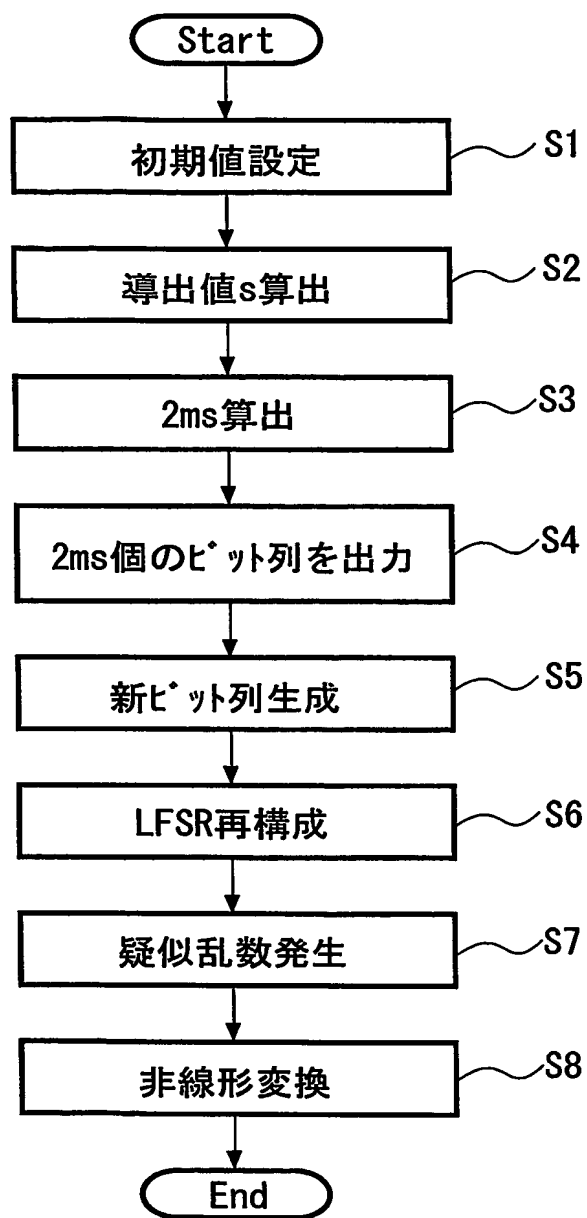


第2図

LSFR1	$x^{131} + x^8 + x^3 + x^2 + 1$
LSFR2	$x^{137} + x^{21} + 1$
LSFR3	$x^{139} + x^8 + x^5 + x^3 + 1$
LSFR4	$x^{149} + x^{10} + x^9 + x^7 + 1$
LSFR5	$x^{151} + x^3 + 1$
LSFR6	$x^{157} + x^6 + x^5 + x^2 + 1$
LSFR7	$x^{163} + x^7 + x^6 + x^3 + 1$
LSFR8	$x^{167} + x^6 + 1$

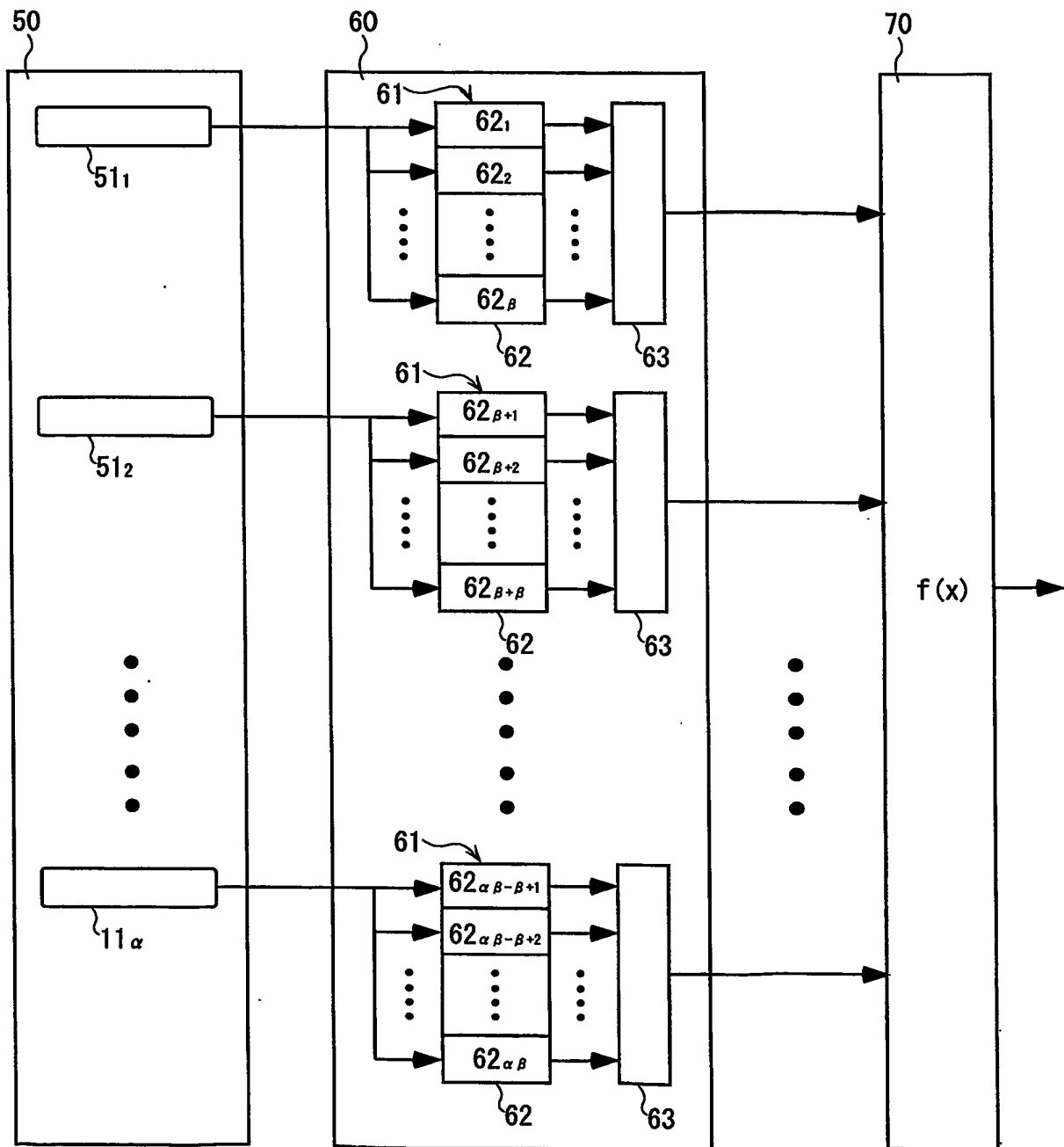
2/15

第3図



3/15

第4図



4/15

第5図

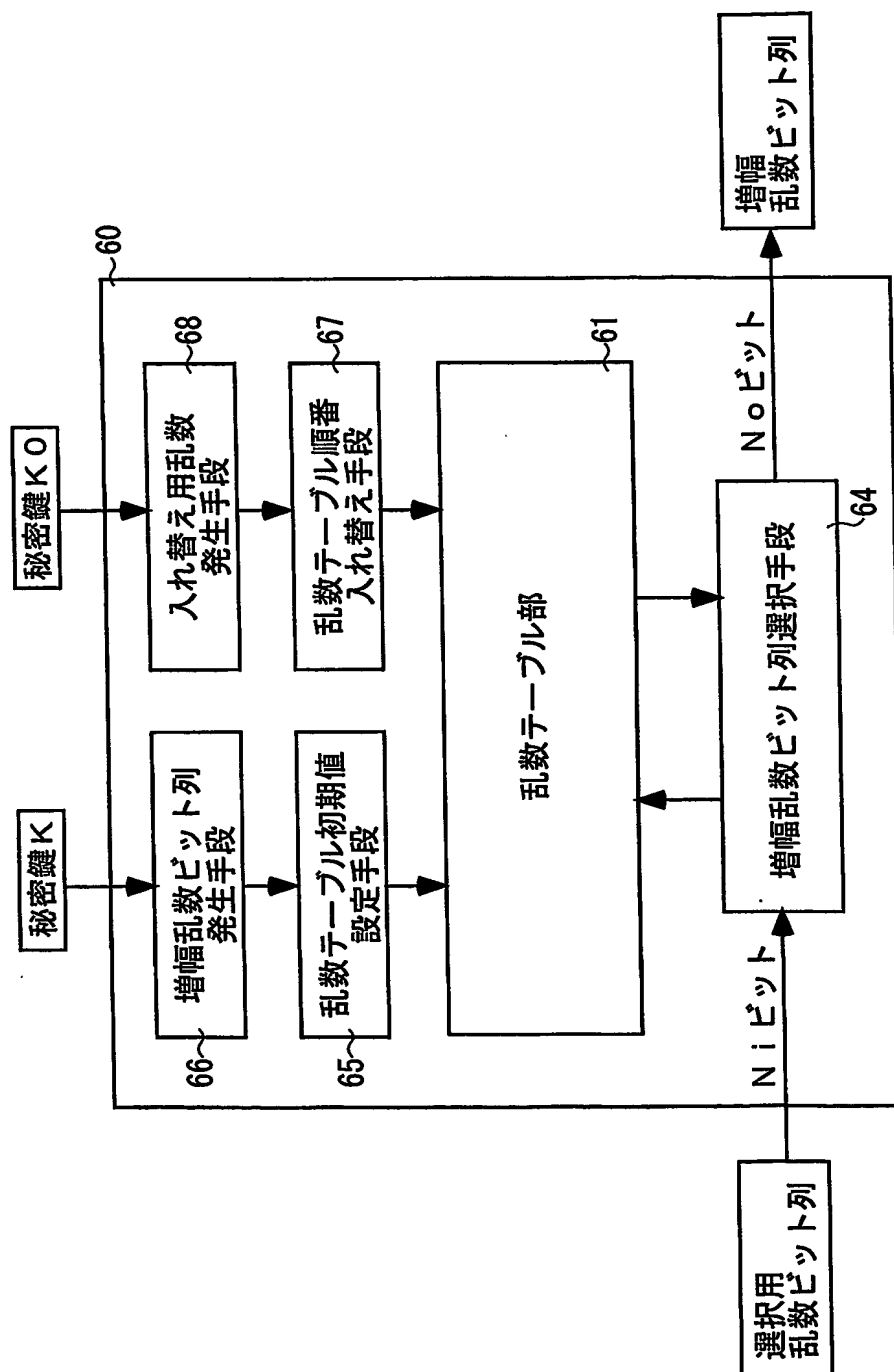
Ri		Ro	
000	01001110010100100010100	
001	00101001101000111001010	
002	00011100101000011011110	
003	01001100011110000101111	
⋮		⋮	

2^{Ni} 個

Niビット Noビット

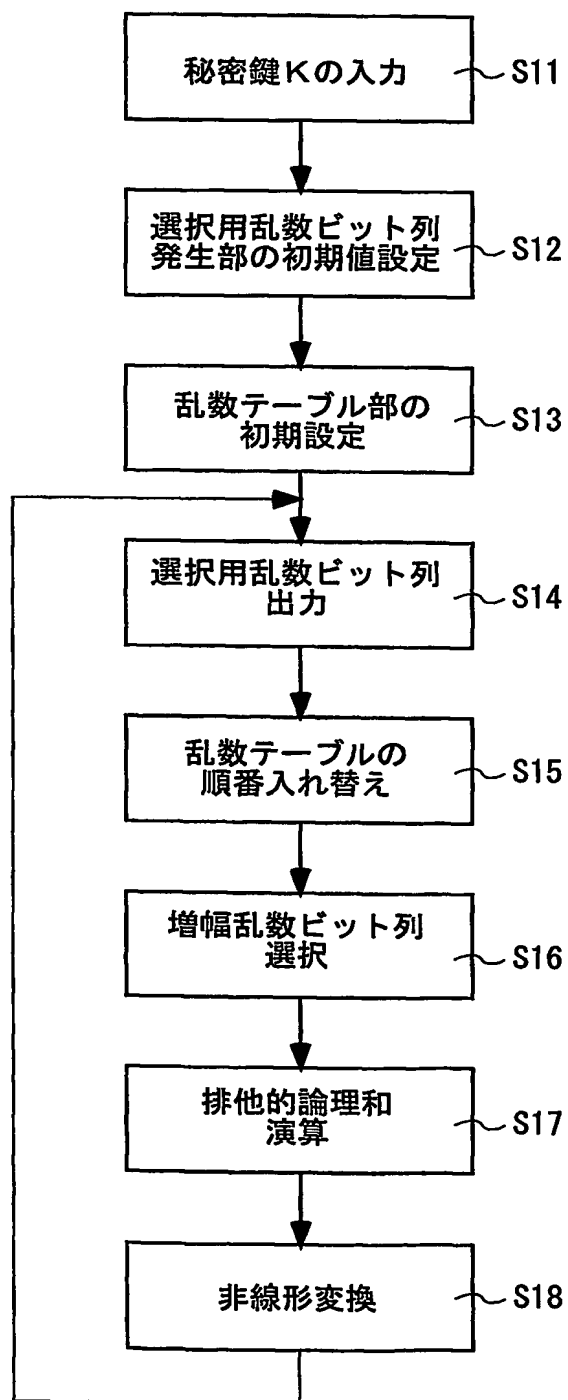
5/15

第6図



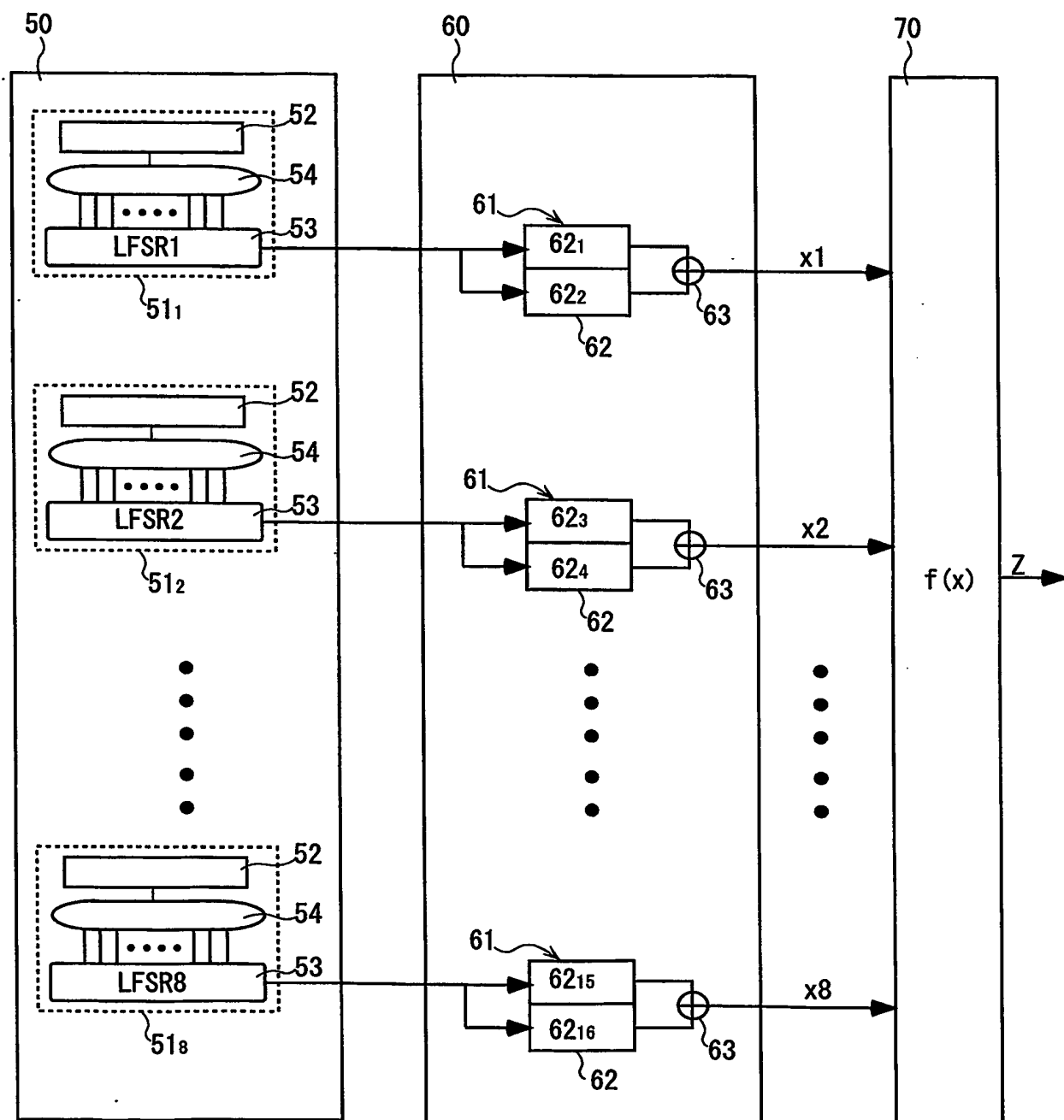
6/15

第7図



7/15

第 8 図



第9図

 $\alpha = 8, \beta = 2, N_i = 2^{16}, L_k = 128$ の場合

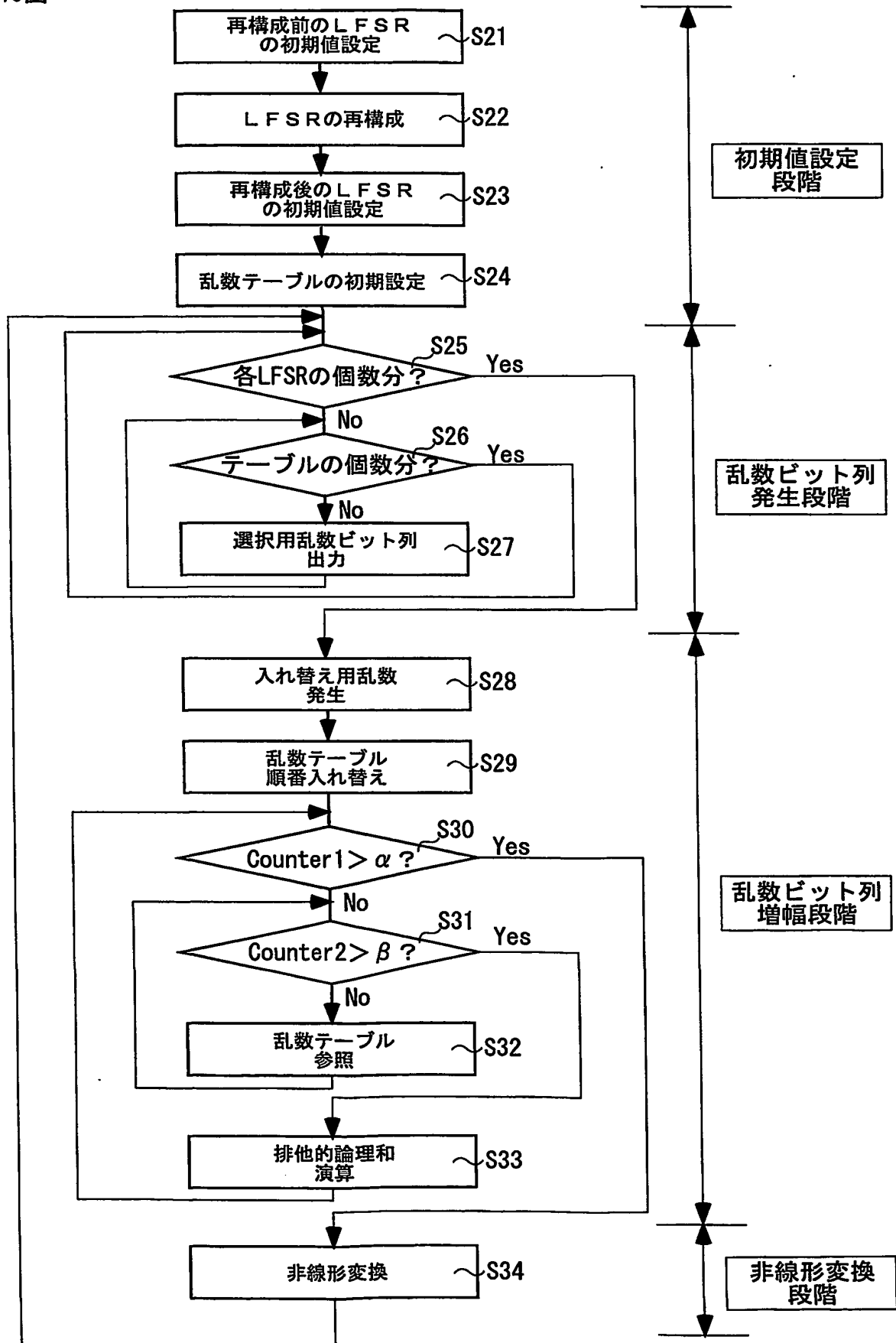
 選択用乱数ビット列
出力手段61₁に対応

 選択用乱数ビット列
出力手段61₂に対応

 選択用乱数ビット列
出力手段61₃に対応

Ri	Ro
0	010110101100010110
1	101101011000101100
2	101011010110001011
3	010110101101110110
.	.
.	.
.	.
255	010110100010101010
0	001010101100010110
1	101011101000101100
2	000101101011010111
3	011011010110101101
.	.
.	.
.	.
255	001010110101100110
0	101010010010101010
1	101101011000101110
2	111010100110001011
3	010110110101011011
.	.
.	.
.	.
255	110101101000010110
0	100010110010110101
1	001011100010110101
2	010110101100101011
3	001101011010110111
.	.
.	.
.	.
255	110100101101010010
	↙
0	010011011010110001
1	010110110101100100
2	110001011010110110
3	011011100111010110
.	.
.	.
.	.
255	010101101100011001
0	110110110111001010
1	100110110111011100
2	011010101110011010
3	011011010111011001
.	.
.	.
.	.
255	010101010110001101

第10図

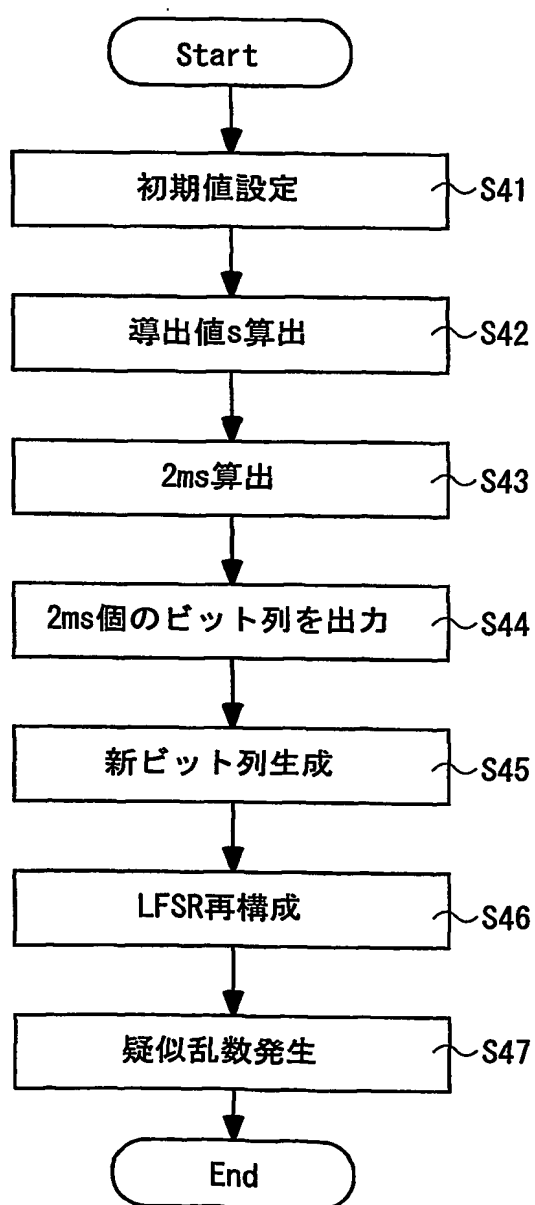


第11図

LFSR1	$I1(x) = x^{129} + x^{80} + x^8 + x + 1$
LFSR2	$I2(x) = x^{131} + x^{80} + x^{16} + x + 1$
LFSR3	$I3(x) = x^{133} + x^{16} + x^8 + x^2 + 1$
LFSR4	$I4(x) = x^{137} + x^{20} + x^{12} + x + 1$
LFSR5	$I5(x) = x^{139} + x^{80} + x^{12} + x + 1$
LFSR6	$I6(x) = x^{143} + x^{56} + x^{12} + x + 1$
LFSR7	$I7(x) = x^{149} + x^{84} + x^8 + x^2 + 1$
LFSR8	$I8(x) = x^{151} + x^{60} + x^8 + x + 1$

11/15

第12図



12/15

第13図

表1:スループット計測結果

LFSR(151段)の平均スループット	5.203Mbits/sec
RC4の平均スループット	141.6Mbits/sec
入れ替え処理の平均スループット	69.99Mbits/sec
乱数テーブルの平均スループット	7.923Gbits/sec
非線形関数の平均スループット	119.2Mbits/sec
疑似乱数発生器全体のスループット	116.4Mbits/sec
従来型の平均スループット	0.680Mbits/sec

第14図

表2:NISTのパラメータ

Data Length	1,000,000
Block Frequency Test Block Length	100
Non Overlapping Template Test Block Length	10
Overlapping Template Test Block Length	10
Universal Test Block Length	7
Universal Test Number Of Initialization Steps	1,280
Approximate Entropy Test Block Length	2
Serial Test Block Length	2
Linear Complexity Test Subsequence Length	500
bitstreams should be generated	10

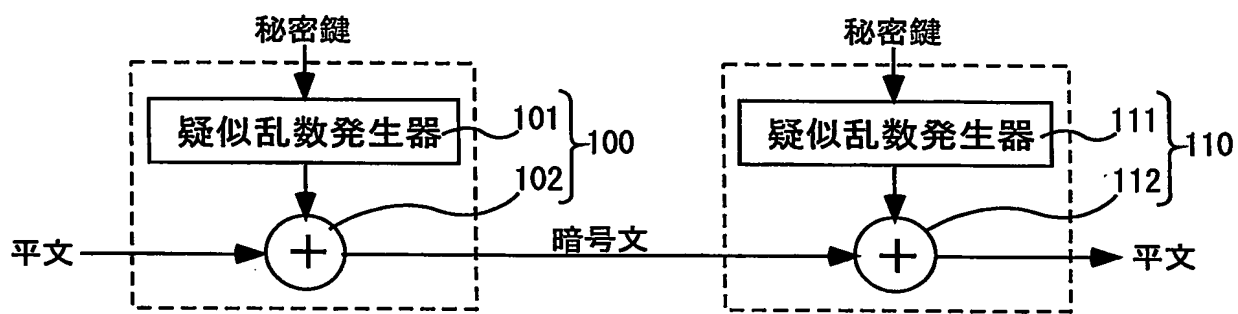
13/15

第15図

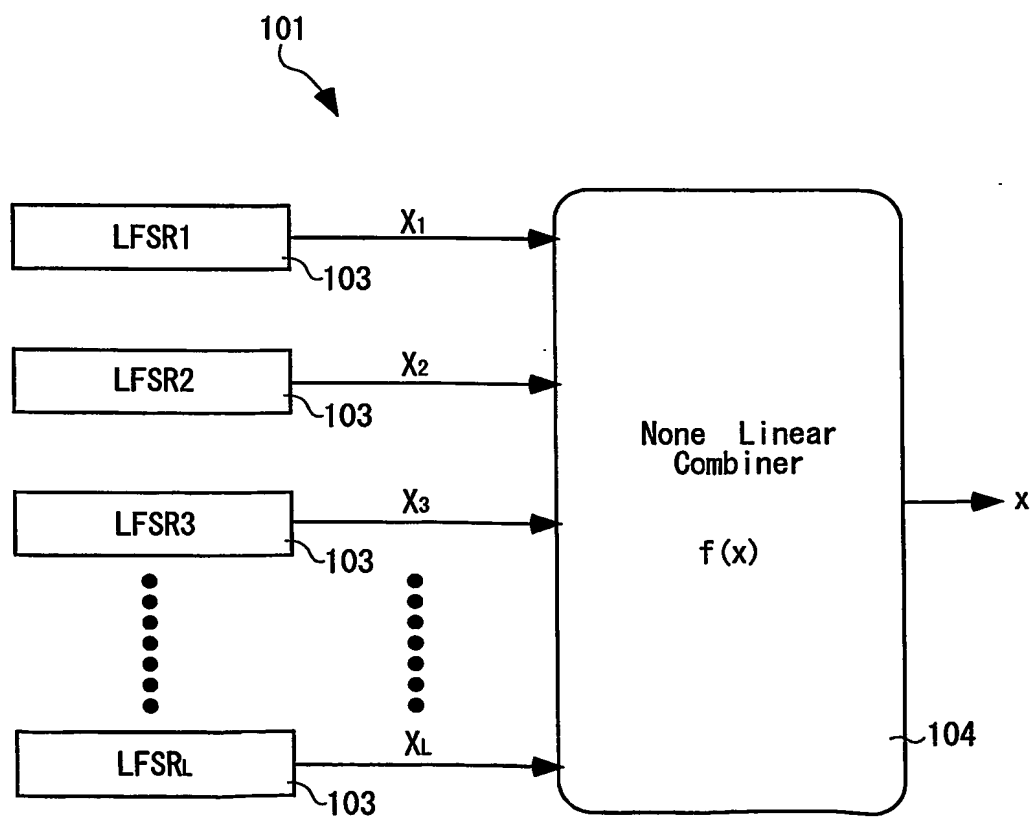
テスト項目	P-value	テスト項目	P-value	テスト項目	P-value	テスト項目	P-value
Frequency	0.987896	Aperiodic-Template	0.657933	Aperiodic-Template	0.616305	Random-Excursion-V	0.035174
Block-Frequency	0.678686	Aperiodic-Template	0.494392	Aperiodic-Template	0.042808	Random-Excursion-V	0.911413
Cusum	0.366918	Aperiodic-Template	0.987896	Aperiodic-Template	0.759756	Random-Excursion-V	0.350485
Cusum	0.213309	Aperiodic-Template	0.419021	Aperiodic-Template	0.779188	Random-Excursion-V	0.534146
Runs	0.383827	Aperiodic-Template	0.350485	Aperiodic-Template	0.262249	Random-Excursion-V	0.350485
Long-Run	0.595549	Aperiodic-Template	0.616305	Aperiodic-Template	0.455937	Random-Excursion-V	0.213309
Rank	0.798139	Aperiodic-Template	0.935718	Aperiodic-Template	0.55442	Random-Excursion-V	0.213309
FFT	0.021999	Aperiodic-Template	0.897763	Aperiodic-Template	0.383827	Random-Excursion-V	0.534146
Aperiodic-Template	0.867692	Aperiodic-Template	0.897763	Aperiodic-Template	0.12962	Random-Excursion-V	0.534146
Aperiodic-Template	0.574903	Aperiodic-Template	0.171867	Aperiodic-Template	0.867692	Random-Excursion-V	0.122325
Aperiodic-Template	0.779188	Aperiodic-Template	0.739918	Aperiodic-Template	0.759756	Random-Excursion-V	0.213309
Aperiodic-Template	0.350485	Aperiodic-Template	0.897763	Aperiodic-Template	0.637119	Random-Excursion-V	0.066882
Aperiodic-Template	0.798139	Aperiodic-Template	0.224821	Aperiodic-Template	0.867692	Random-Excursion-V	0.035174
Aperiodic-Template	0.494392	Aperiodic-Template	0.851383	Aperiodic-Template	0.955835	Serial	0.350485
Aperiodic-Template	0.867692	Aperiodic-Template	0.419021	Aperiodic-Template	0.897763	Serial	0.289667
Aperiodic-Template	0.085587	Aperiodic-Template	0.319084	Aperiodic-Template	0.996335	Lempel-Ziv	0.383827
Aperiodic-Template	0.474986	Aperiodic-Template	0.401199	Aperiodic-Template	0.115387	Linear-Complexity	0.090936
Aperiodic-Template	0.996335	Aperiodic-Template	0.616305	Aperiodic-Template	0.383827		
Aperiodic-Template	0.249284	Aperiodic-Template	0.911413	Aperiodic-Template	0.275709		
Aperiodic-Template	0.153763	Aperiodic-Template	0.897763	Aperiodic-Template	0.55442		
Aperiodic-Template	0.514124	Aperiodic-Template	0.897763	Aperiodic-Template	0.051942		
Aperiodic-Template	0.657933	Aperiodic-Template	0.897763	Aperiodic-Template	0.595549		
Aperiodic-Template	0.595549	Aperiodic-Template	0.080519	Aperiodic-Template	0.657933		
Aperiodic-Template	0.719747	Aperiodic-Template	0.867692	Aperiodic-Template	0.637119		
Aperiodic-Template	0.996335	Aperiodic-Template	0.115387	Aperiodic-Template	0.045675		
Aperiodic-Template	0.657933	Aperiodic-Template	0.275709	Aperiodic-Template	0.924076		
Aperiodic-Template	0.759756	Aperiodic-Template	0.779188	Aperiodic-Template	0.978072		
Aperiodic-Template	0.834308	Aperiodic-Template	0.202268	Aperiodic-Template	0.739918		
Aperiodic-Template	0.851383	Aperiodic-Template	0.319084	Aperiodic-Template	0.455937		
Aperiodic-Template	0.657933	Aperiodic-Template	0.637119	Aperiodic-Template	0.657933		
Aperiodic-Template	0.494392	Aperiodic-Template	0.739918	Aperiodic-Template	0.574903		
Aperiodic-Template	0.779188	Aperiodic-Template	0.224821	Aperiodic-Template	0.304126		
Aperiodic-Template	0.883171	Aperiodic-Template	0.514124	Aperiodic-Template	0.249284		
Aperiodic-Template	0.798139	Aperiodic-Template	0.137282	Aperiodic-Template	0.289667		
Aperiodic-Template	0.719747	Aperiodic-Template	0.964295	Aperiodic-Template	0.946308		
Aperiodic-Template	0.964295	Aperiodic-Template	0.334538	Aperiodic-Template	0.010535		
Aperiodic-Template	0.401199	Aperiodic-Template	0.678686	Aperiodic-Template	0.816537		
Aperiodic-Template	0.12962	Aperiodic-Template	0.719747	Aperiodic-Template	0.739918		
Aperiodic-Template	0.739918	Aperiodic-Template	0.080519	Aperiodic-Template	0.350485		
Aperiodic-Template	0.010555	Aperiodic-Template	0.145326	Aperiodic-Template	0.798139		
Aperiodic-Template	0.202268	Aperiodic-Template	0.319084	Aperiodic-Template	0.455937		
Aperiodic-Template	0.289667	Aperiodic-Template	0.145326	Aperiodic-Template	0.145326		
Aperiodic-Template	0.897763	Aperiodic-Template	0.304126	Periodic-Template	0.657933		
Aperiodic-Template	0.719747	Aperiodic-Template	0.867692	Universal	0.383827		
Aperiodic-Template	0.494392	Aperiodic-Template	0.719747	Apen	0.534146		
Aperiodic-Template	0.019188	Aperiodic-Template	0.437274	Random-Excursion	0.534146		
Aperiodic-Template	0.066882	Aperiodic-Template	0.030806	Random-Excursion	0.213309		
Aperiodic-Template	0.574903	Aperiodic-Template	0.224821	Random-Excursion	0.534146		
Aperiodic-Template	0.699313	Aperiodic-Template	0.514124	Random-Excursion	0.035174		
Aperiodic-Template	0.978072	Aperiodic-Template	0.171867	Random-Excursion	0.534146		
Aperiodic-Template	0.153763	Aperiodic-Template	0.010988	Random-Excursion	0.534146		
Aperiodic-Template	0.419021	Aperiodic-Template	0.946308	Random-Excursion	0.010879		
Aperiodic-Template	0.851383	Aperiodic-Template	0.162606	Random-Excursion	0.213309		
Aperiodic-Template	0.55442	Aperiodic-Template	0.534146	Random-Excursion-V	0.534146		
Aperiodic-Template	0.897763	Aperiodic-Template	0.574903	Random-Excursion-V	0.739918		
Aperiodic-Template	0.213309	Aperiodic-Template	0.334538	Random-Excursion-V	0.213309		
Aperiodic-Template	0.319084	Aperiodic-Template	0.699313	Random-Excursion-V	0.911413		

14/15

第16図

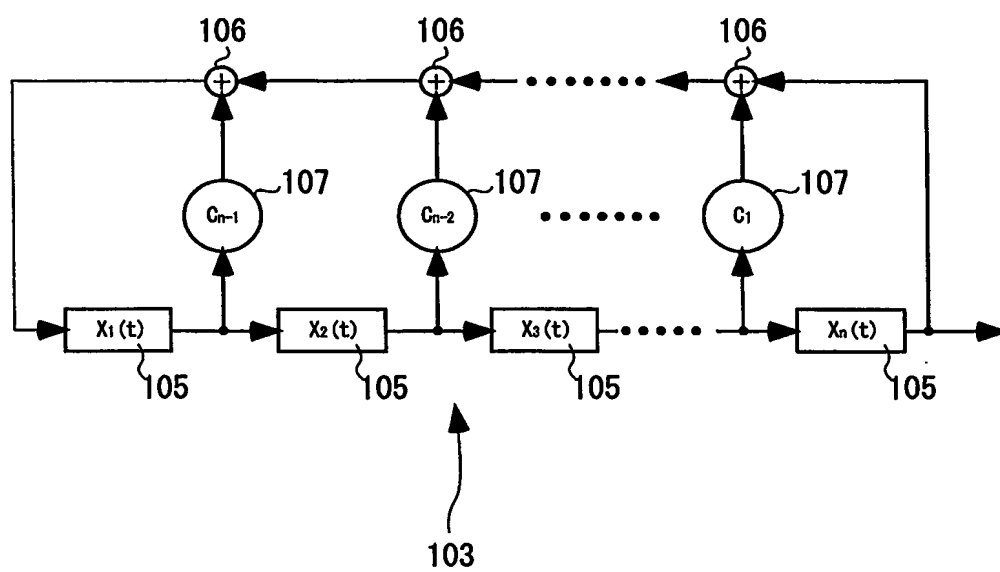


第17図



15/15

第18図



第Ⅷ欄 (v) 不利にならない開示又は新規性喪失の例外に関する申立て

申立ては実施細則第 215 号に規定する標準文言を使用して作成しなければならない。第Ⅷ欄と同欄(i)～(v)の備考の総論部分、及び本頁に特有の事項について第Ⅷ欄(v)の備考を参照。この欄を使用しないときは、この用紙を願書に含めないこと。

不利にならない開示又は新規性喪失の例外に関する申立て (規則 4.17(v)及び 51 の 2.1(a)(v))

本国際出願 [PCT/JP03/08794] に関し、

小林 朗、森井 昌克、白石 善明は、本国際出願の請求項に記載された対象が以下のように開示されたことを申し立てる。

- (i) 開示の種類：刊行物
- (ii) 開示の日付：2003年1月26日
- (iii) 開示の名称：2003年 暗号と情報セキュリティシンポジウム予稿集Vol.II
- (iv) 開示の場所：静岡県浜松市板屋町 アクトシティ浜松 コンgressセンター
- (v) 本申立ては、すべての指定国のためになされたものである。



この申立ての続業として「第Ⅷ欄(v)の続き」がある

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/08794

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ G09C1/00, G06F7/58

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G09C1/00, G06F7/58

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Jitsuyo Shinan Toroku Koho	1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 10-240500 A (Toshiba Corp.), 11 September, 1998 (11.09.98), Figs. 3 to 12 (Family: none)	7-10, 14-17 11-13, 18-20
X	JP 9-179726 A (NEC Corp.), 11 July, 1997 (11.07.97), Full text & EP 782069 A	7, 9, 14, 16
X	JP 62-144243 A (NEC Corp.), 27 June, 1987 (27.06.87), Full text (Family: none)	7, 9, 14, 16

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
 "A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

Date of the actual completion of the international search
26 August, 2003 (26.08.03)

Date of mailing of the international search report
09 September, 2003 (09.09.03)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/08794

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 4-115616 A (Toshiba Corp.), 16 April, 1992 (16.04.92), Full text (Family: none)	11-13, 18-20
A	JP 7-36672 A (Canon Inc.), 07 February, 1995 (07.02.95), Full text & EP 635956 A	1-20
A	JP 7-104976 A (NEC Corp.), 21 April, 1995 (21.04.95), Full text & US 5566099 A	1-20
A	JP 61-246787 A (Fujitsu Ltd.), 04 November, 1986 (04.11.86), Full text (Family: none)	7-20

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl.⁷ G09C 1/00 , G06F 7/58

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl.⁷ G09C 1/00 , G06F 7/58

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年
 日本国公開実用新案公報 1971-2003年
 日本国登録実用新案公報 1994-2003年
 日本国実用新案登録公報 1996-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	J P 10-240500 A (株式会社東芝) 1998. 09. 11, 第3図-第12図 (ファミリーなし)	7-10, 14-17
Y		11-13, 18-20
X	J P 9-179726 A (日本電気株式会社) 1997. 07. 11, 全文 & EP 782069 A	7, 9, 14, 16

☒ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
 「O」 口頭による開示、使用、展示等に言及する文献
 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」 同一パテントファミリー文献

国際調査を完了した日

26. 08. 03

国際調査報告の発送日

09.09.03

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)
 郵便番号100-8915
 東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

石田 信行



5M

9469

電話番号 03-3581-1101 内線 3598

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	JP 62-144243 A (日本電気株式会社) 1987. 06. 27, 全文 (ファミリーなし)	7, 9, 14, 16
Y	JP 4-115616 A (株式会社東芝) 1992. 04. 16, 全文 (ファミリーなし)	11-13, 18-20
A	JP 7-36672 A (キヤノン株式会社) 1995. 02. 07, 全文 & EP 635956 A	1-20
A	JP 7-104976 A (日本電気株式会社) 1995. 04. 21, 全文 & US 5566099 A	1-20
A	JP 61-246787 A (富士通株式会社) 1986. 11. 04, 全文 (ファミリーなし)	7-20